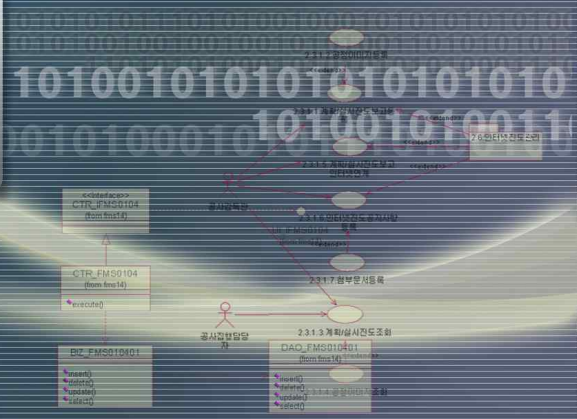


국방 CBD 방법론 V2.0

사용자 지침서 1권

Component
Based
Development



**MND
COMPONENT
BASED
DEVELOPMENT
METHODOLOGY**

V2.0 User Guide Volume 1



국 방 부

머 리 말

국방정보화는 국방개혁의 원동력으로, 우리 군은 미래 전장개념인 네트워크 중심전(NCW) 환경을 만들어 나가기 위해 전장 및 자원관리정보시스템을 다양한 임무 기능에 맞추어 구축·운용하고 있으며, 군 운용환경의 변화에 대응하여 고도화를 추진해 나가고 있습니다.

‘05년 4월부터 적용되고 있는 국방 CBD 방법론(v1.1)은 개발 공정 및 산출물의 표준화, 상호운용성 증진, 공통 컴포넌트 개발 및 재활용 등에 많은 기여를 하였으나 국방 내·외부 정보화 환경의 변화, 정보기술의 발전 등으로 인하여 지속적으로 개선이 요구되어 왔습니다. 이에 전문기관의 연구와 국방부, 합참, 방위사업청 등의 의견 수렴, 외부 전문가 자문을 통해 국방 CBD 방법론(v2.0)을 제정·시행하게 되었습니다.

국방 CBD 방법론(v2.0)은 기존 방법론 자체의 완전성을 제고하기 위하여 불필요 산출물 삭제, 필요 산출물 추가, 상호운용성 관련 절차(SHADE, COE, MND-AF)와의 연계성 확보, 최신 기술 동향 반영 등의 사항을 중점으로 작성되었으며, 소규모 사업의 범위를 확대하고 사업의 성격에 따라 유연하게 테일러링할 수 있는 가이드를 보완하여 산출물 작성 부담을 경감할 수 있도록 하였습니다.

개선된 국방 CBD 방법론을 적용하여 정보시스템의 품질 향상과 내부 개발 프로세스를 견고히 하고, 공통 컴포넌트 개발 및 활용을 통한 경제적이고 효율적인 정보화 환경을 구현하여 국방정보화의 소프트 인프라웨어 확산 전략과 국방 공통운용환경 구축을 위한 기틀이 될 수 있기를 바라며, 향후 지속적인 개선이 이루어질 수 있도록 관련자들의 적극적인 관심과 활발한 참여를 당부합니다.

2010년 5월 7일

김 재 민

국 방 부 정 보 화 기 획 관 고 위 공 무 원 김 재 민

국방 CBD 방법론 v2.0

(사용자 지침서 1권)



1. 개 요
2. 세부절차
3. 테일러링 가이드
4. 절차 및 서식 설명
5. 국방정보기술아키텍처 활용 가이드

Synopsis

개 요

개요

■ 제정 목적

국방 컴포넌트 기반 개발 방법론(이하 국방 CBD 방법론)은 UML(Unified Modeling Language) 기반의 CBD(Component Based Development) 방법론에 기초하여 분석, 설계 및 구현 산출물을 표준화시켜 여러 사람이 설계하고 개발하더라도 서로 간의 의사소통이 쉽고, 일관성을 유지하며, 최소의 노력으로 높은 품질의 컴포넌트 기반 정보시스템을 개발하기 위해 제정되었다.

(관련 근거 : 기반체계담당관-928, '국방 아키텍처 프레임워크(MND-AF)와 CBD방법론 적용 지시', 2005. 4. 7)

■ 개정 목적

'05년 4월부터 국방 CBD 방법론(ADDMe v1.1)을 국방 정보체계 개발 사업에 표준 방법론으로 적용한 결과 산출물 작성이 과다하고, 유사항목이 중복 작성되는 문제점이 도출되었고 COE, SHADE 및 국방 아키텍처 프레임워크의 적용으로 인해 방법론과 중복되는 부분이 발생하였으며, 신기술의 반영에 대한 요구가 증대됨에 따라 본 방법론을 개정하게 되었다.

(관련 근거 : 정보화정책담당관-1663, '국방 CBD 방법론(v2.0) 적용 지시', 2010. 5. 7)

■ 주요 개정 내용

국방 CBD 방법론이 v2.0으로 개정되면서 개선된 내용은 다음과 같다.

- 국방 CBD 방법론 v1.1 산출물 내의 중복 항목 최소화, 유사 산출물 통합 등을 통하여 산출물 작성 부담을 줄이고 중복으로 인한 내용 불일치 가능성을 배제하였으며 산출물 관리 용이성을 제고하였다.
- 국방 아키텍처 프레임워크(MND-AF) v1.2와 유사한 항목에 대해 서식을 일치함으로써 현행 아키텍처를 참조하거나 목표 아키텍처로 재활용이 가능하도록 하여 산출물 작성에 있어 사용자 편의성을 제고하였다.
- COE, SHADE 등 국방 정보체계 개발 시 준수해야 할 지침을 작업 방법에 포함하여 국방 정책 방향에 맞게 시스템이 개발되도록 하였다.
- 정보체계 개발 사업에 따라 필요한 데이터 구축, 연동 정의 및 구현, 보안 요소 적용 등에 관한 작업 및 산출물을 추가하였다.
- 기존 테일러링 가이드의 정보체계 규모 기준을 개선하고 유지보수 사업, 패키지 개발 등에 관한 가이드를 포함하였다.
- 최신 정보 기술 기법 중 국방 정보체계 개발에 적용 가능한 요소를 선별하여 작업 설명과 기법에 반영하였다.
- 국방 CBD 방법론의 적용 범위를 명확하게 하고 국방 정보체계 사업관리 규정에서 정의하고 있는 표준 공정 간의 연관성을 제시하였다

개요

적용 범위

국방 CBD 방법론은 전장관리정보체계(무기체계)와 자원관리정보체계(비무기체계) 등의 소프트웨어를 컴포넌트 기반으로 개발 시 적용하며, 적용 방법 및 근거는 국방전력발전업무훈령과 국방정보체계사업관리지시 등 관련 규정에 따른다.

정보체계 개발 관련 프로세스에서 적용 범위는 다음과 같다.

『소프트웨어사업 관리감독에 관한 일반기준 (정보통신부 고시 제2006-39호)』의 「공공부문 SW사업 발주·관리 표준 프로세스 프레임워크」에 의하면 수명주기 프로세스는 크게 핵심(Primary), 지원(Supporting), 조직(Organizational)로 나뉘며 각각의 수명 주기 프로세스는 프로세스, 활동, 작업으로 세분화된다.

전체 프로세스에 대한 세부 규칙은 국방전력발전업무규정, 국방정보체계사업관리지시 등 관련 규정에서 명시한다.

국방 CBD 방법론은 핵심 수명주기 프로세스에서 개발(development) 프로세스를 컴포넌트 기반 개발 방법을 적용하였을 경우 이에 대한 개발 공정 절차와 산출물을 정의한다.

이외 지원 및 조직 수명주기 프로세스는 국방 CBD 방법론에서 다루지 않으므로 개발 조직(제안 업체)이 보유한 고유의 방법론을 국방 CBD 방법론과 상충되지 않도록 보완하여 적용해야 한다.



■ 방법론 구성 배경 및 원칙

국방 CBD 방법론은 국방 정보화 환경 내에서 컴포넌트 기반의 표준 개발 방법론으로써 다음과 같은 국방 정보체계 개발 배경을 고려하여 구성되었다.

■ 다양한 도메인 존재

국방 정보체계는 크게 전장, 자원 도메인 내에서 지휘통제, 군사정보, M&S, 기획·재정, 군수·시설, 인사·동원 등 다양한 세부 도메인이 존재한다.

국방 CBD 방법론은 각 도메인에 적용이 가능한 범용적 특성을 가진다.

그러므로 모든 정보체계 개발에 필요한 공정과 산출물을 모두 정의하지 아니하고 정보체계 개발 사례 분석을 통해 추출된 공통적 개발 방법을 정의한다.

■ 다양한 사업 케이스 존재

국방 정보체계 개발 사업은 대규모, 소규모, 긴급, 정부 시책, 진화 또는 일괄적 개발, 상용 패키지 도입 등 다양한 사업 케이스가 존재한다.

국방 CBD 방법론은 각 사업 케이스에 적용이 가능한 범용적 특성을 가진다.

그러므로 이 모든 경우에 대한 세부 지침을 열거하지 아니하고 테일러링 가이드를 통해 각 사업별로 특성에 맞는 방법론이 될 수 있도록 유도한다.

또한 특정 상용 제품이 도입될 경우 수반되는 프로세스 및 산출물의 추가 또는 변형에 대한 나열식 정의를 생략하고 해당 사업의 특성에 맞도록 추가 공정을 수행하고 산출물을 첨부하도록 유도한다.

■ 다양한 개발 조직 참여

국방 정보체계 개발 사업은 대기업, 중소기업, 자체 개발팀 등 다양한 개발 조직이 참여하고 각 개발 조직은 다양한 소프트웨어 개발 성숙도를 보유하고 있으며, 또한 각자 고유의 개발 방법론을 보유하고 있다.

따라서 국방 CBD 방법론은 표준 개발 방법론으로서 이해하기 쉽게 구성되고 성숙도가 높은 기술 요소와 일반적으로 널리 알려진 기법을 적용한다.

또한 테일러링 가이드를 통하여 각 개발 조직 고유의 방법론과의 융합이 가능토록 한다.

■ 별도 규정과 지침 존재

국방 정보화 환경은 국방전력발전업무훈령, 국방정보체계사업관리지시, 국방상호 운용성관리훈령과 같은 상위 규정과 국방 아키텍처 프레임워크 적용 지시, 데이터 표준화 지침과 같은 세부 수행 지침이 존재한다.

따라서 국방 CBD 방법론은 각 규정 및 지침을 준수하면서 각 규정 및 지침으로부터 방법론의 내용을 분리하여 상위 규정의 개정에 따른 영향을 최소화하도록 한다.

■ 방법론 구성 개념

국방 CBD 방법론의 내용은 국제 표준인 ISO/IEC 12207 - Software Life Cycle Processes의 프로세스를 준수하면서 단지 소프트웨어 개발 방법에 국한하지 않고 정보 시스템의 개발 방법에 중점을 두었다. 정보 시스템은 소프트웨어, 데이터, 하드웨어, 네트워크 등의 통합이라고 정의될 수 있다.

그러나 일반적으로 하드웨어, 네트워크 등은 상용 제품으로 구성되므로 본 방법론은 이에 대한 분석 및 설계, 구현은 제외하며 소프트웨어와 데이터의 분석 및 설계, 구현에 초점을 두고 기타 요소와의 통합에 초점을 둔다.



개요

■ 방법론 구조

국방 CBD 방법론은 프로세스와 관련 문서로 구성되어 있으며, 프로세스 구조는 다음과 같다.



국방 CBD 방법론의 프로세스는 단계, 활동, 작업, 절차로 구성된다. 각 단계는 여러 개의 활동들로, 각 활동은 여러 개의 작업들로, 각 작업은 여러 개의 절차들로 이루어져 있다.

■ 프로세스

프로세스는 4개의 단계, 15개의 활동, 37개의 작업으로 구성된다.

● 단계

프로세스의 최상위 수준으로 활동들의 구조적 집합이다.

● 활동

논리적 연결성이 있는 작업들의 구조적 집합이다.

● 작업

개발자가 수행하는 최소 단위의 일로써 하나 이상의 절차들로 구성된다.

● 절차

프로세스의 최하위 수준으로 작업을 수행하기 위한 순서에 해당한다.

■ 구성 문서

● 테일러링 가이드

방법론의 공정과 산출물을 해당 사업에 맞게 테일러링할 수 있도록 가이드를 제공하고 있다.

● 국방정보기술아키텍처 활용 가이드

국방정보기술아키텍처 구축 결과를 참고하거나 시스템 개발 결과를 기반으로 단위 체계 아키텍처를 구축하는데 도움이 되도록 참고 관계를 제시한다.

● 단계 설명서

해당 단계의 의미와 단계에서 수행할 수 있는 활동 및 작업들의 전반적인 내용을 담고 있다.

● 작업 설명서

해당 작업의 의미와 작업을 구성하는 입력물과 산출물, 참여자, 관련 기법을 명시하여 ICOM(Input, Control, Output, Mechanism)을 파악할 수 있으며, 실제 작업 수행 방법을 이해할 수 있도록 세부 절차를 정의한다.
또한 작업과 산출물이 연결될 수 있도록 산출물 서식과 작성 항목을 기술한다.

● 산출물 예시

산출물의 작성 예시를 통해 작성 방법을 보다 쉽게 이해할 수 있다.

● 기법서

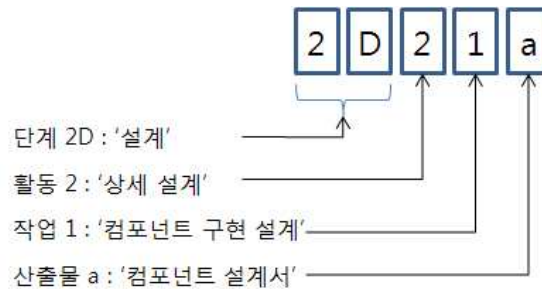
기법서는 작업의 결과를 산출물로 작성하는데 참고하는 기법에 대한 설명을 담고 있다. 해당 기법의 예시를 두어 기법의 이해를 도우며 구성 요소, 지침 및 고려사항을 두어 기법이 설명되어 있다.

개요

방법론의 표준

■ 단계, 활동, 작업, 산출물의 식별부호 체계

식별부호는 두 자리의 단계 식별부호와 한 자리의 활동 일련번호, 한 자리의 작업 일련번호, 한 자리의 산출물 번호로 나타내어진다.



- 단계 식별부호 = 단계 일련번호(단계에 부여한 일련번호) + 단계 약어

단계	분석	설계	구현 및 테스트	인도
단계 약어	R(Requirement)	D(Design)	T(Test)	S(Solution)

- 활동 식별부호 = 단계 식별부호 + 활동 일련번호(활동에 부여한 일련번호)
- 작업 식별부호 = 활동 식별부호 + 작업 일련번호(작업에 부여한 일련번호)
- 산출물 식별부호 = 작업 식별부호 + 산출물번호(a/b/c/...)

■ 산출물 서식 범례

산출물 서식에서 다음과 같은 참조/기재 방법이 존재한다.

● 실제 대상명 기재

산출물 서식에서 '(항목명)'과 같이 괄호내 제목은 서식이 아니라 개발 대상 시스템중 실제 대상의 명칭을 기재해야 한다.

예 : [서식] 1. (컴포넌트명) → [산출물] 1. BE_MDLMgt

● 각주

- 산출물 서식에서 각주 '*'에 해당하는 항목은 국방 아키텍처 프레임워크 (MND-AF) v1.2 유관 항목이므로 연관된 MND-AF 산출물 내용을 참조하여 기재한다. (본 방법론의 「국방정보기술아키텍처 활용 가이드」 참고)
- 각주 '**'에 해당하는 항목은 '데이터 표준화 지침' 유관 항목이므로 해당 지침을 참조하여 작업을 수행하고 산출물을 작성한다.

■ 산출물 작성 방법

국방 CBD 방법론의 산출물을 작성할 시 다음과 같은 작성 방법을 고려한다.

- 산출물 서식에는 산출물을 구성하는 필수 항목에 대해서만 제공한다. 따라서 산출물 작성 시에는 각 프로젝트에서 정의한 문서 표준에 맞게 표지, 경고문, 작성 이력 등을 포함해서 작성해야 한다.

<작성 이력 예시>

날짜	버전	변경자	내용	확인자

- 부록은 지속적인 관리가 용이하도록 산출물에서 분리하여 구성한 보조 산출물로써 해당 작업을 수행한다면 반드시 작성해야 한다.
- 두 작업에서 작성하는 일부 산출물의 경우 선행 작업에서 산출물의 일부를 작성하고 후행 작업에서 나머지를 작성하여 완성한다.
- 방법론에서 산출물 서식을 워드프로세서 형태로 제공하고 있지만 스프레드시트로 작성하는 것이 용이한 경우 스프레드시트 문서로 작성/관리/제출이 가능하다.
- 대규모 시스템의 경우 각 산출물에 대해 하나로 작성 및 관리하기 어려우므로 소프트웨어 구성 항목(CSCI : Computer Software Configuration Item) 또는 서브시스템 등의 단위로 산출물을 분리하여 작성 가능하다.

■ 상호운용성 기반 기술과의 관계

국방 정보체계는 정보유통, 중복개발 방지, 재사용성 제고 등을 위하여 상호운용성을 보장하여야 한다. 이에 따라 국방 CBD 방법론 v2.0에서는 상호운용성에 관련된 기반 기술들이 적용될 수 있도록 방법론의 각 작업에 해당되는 내용을 반영하였으며 체계 개발 시 이를 참조하여 적용해야 한다.

■ 국방 아키텍처 프레임워크(MND-AF)

- 국방 상호운용성 관리 훈령(국방부 훈령 제1110호)에 따라 체계 개발 시 국방 아키텍처 프레임워크(MND-AF)를 적용해야 한다.
- 국방 CBD 방법론 v2.0에서는 MND-AF와의 항목 비교를 통해 유사한 항목에 대해 서식을 일치하였다.
- 이를 통해 시스템 개발 시 국방정보기술아키텍처 구축 결과를 참고하거나 시스템 개발 결과를 기반으로 단위 체계 아키텍처를 구축하는데 도움이 되도록 하였으며 관련 사항을 각 작업 설명 이외에 ‘국방정보기술아키텍처 활용 가이드’로 별도 제시하였으므로 이를 참조해야 한다.

■ 국방 공통운용환경(COE)

- 국방정보체계 개발 시 공통운용환경 하에 관리되고 있는 국방 공통 컴포넌트의 재사용을 우선적으로 검토하여야 한다.
- 이에 따라 2D11 컴포넌트 식별 작업에서 재사용 가능한 컴포넌트의 활용 가능성을 판단한다. 재사용 가능한 컴포넌트가 존재하는 경우, 재사용 컴포넌트와의 연계 및 활용 방법을 정의한다.
- 국방 공통컴포넌트 활용 시 시스템 아키텍처의 변경 소요가 발생할 수 있으므로 이에 대한 검토가 필요하다.
(‘10. 3. 1. 현재 전자정부 표준 프레임워크 기반의 국방 공통 컴포넌트가 다수 존재)
- 체계 개발을 통해 구현된 컴포넌트 중 공통 컴포넌트로 등록하고자 하는 경우 ‘국방 상호운용성 관리 지시’에 따라 등록을 요청한다.

■ 국방 데이터공유환경(SHADE)

- 국방 데이터공유환경(SHADE)의 표준 데이터와 데이터 모델 등은 체계 개발 시 우선적으로 적용해야 하며 신규 표준 데이터는 데이터 등록 절차에 따라 등록하여야 한다. 표준 데이터와 데이터 모델 등은 국방 메타데이터 관리 시스템에서 관리된다.

- 따라서 2D15 데이터 모델링과 2D23 데이터베이스 설계 작업 수행 시 국방 메타데이터 관리 시스템에서 제공하는 표준단어, 도메인, 표준코드, 명명규칙을 반드시 준수해야 하며 데이터 모델은 참조하여야 한다.

■ 국방정보기술표준(DITA)

- 정보체계 개발 시 국방정보기술표준(DITA)을 우선적으로 적용하여야 한다. 따라서 개발 시 필요한 정보기술이 국방 표준에 해당되는지에 대한 검토가 필요하다.
- 이에 따라 1R21 시스템 아키텍처 정의 작업 시 개발 대상 시스템이 어떠한 기술로써 구현되는지를 기술하여야 하며, 해당 기술이 국방 정보기술표준에 등록되지 않은 경우 ‘국방 상호운용성 관리 훈령’에 따라 등록 승인을 요청하여야 한다.

개요

국방 컴포넌트 기반 개발 방법론 개발절차

단계	활동	작업	산출물	부록
1R 분석	1R1 요구사항 식별	1R11 도메인 모델링	1R11a 도메인 정의서 1R11b 용어집	
		1R12 현행 시스템 분석	1R12a 현행 시스템 분석서	
		1R13 비즈니스 모델링	1R13a 비즈니스 정의서	
		1R14 요구사항 정의	1R14a 요구사항 정의서	요구사항 추적표
	1R2 아키텍처 정의	1R21 시스템 아키텍처 정의	1R21a 시스템 아키텍처 정의서	
		1R22 표준 지침 수립	1R22a 표준 지침서	
	1R3 요구사항 분석	1R31 유스케이스 모델링	1R31a 유스케이스 정의서	
		1R32 연동 소요 분석	1R32a 연동 소요 정의서	연동 항목 소요서(IER)
		1R33 사용자 인터페이스 정의	1R33a 사용자 인터페이스 정의서	
		1R34 클래스 모델링	1R34a 클래스 정의서	
	1R4 자료 구축 준비	1R41 자료 구축 계획	1R41a 자료 구축 계획서	
	1R5 테스트 준비	1R51 테스트 계획	1R51a 테스트 계획서	
2D 설계	2D1 개략 설계	2D11 컴포넌트 식별	2D11a 컴포넌트 아키텍처 명세서	
		2D12 인터페이스 상호작용 명세	2D12a 인터페이스 상호작용 명세서	
		2D13 컴포넌트 명세	2D13a 컴포넌트 명세서	
		2D14 사용자 인터페이스 명세	2D14a 사용자 인터페이스 명세서	
		2D15 데이터 모델링	2D15a 데이터 명세서	
	2D2 상세 설계	2D21 컴포넌트 구현 설계	2D21a 컴포넌트 설계서 2D21b 트랜잭션 설계서	
		2D22 사용자 인터페이스 구현 설계	2D22a 사용자 인터페이스 설계서	
		2D23 데이터베이스 설계	2D23a 데이터베이스 설계서	코드 설계서
		2D24 연동 설계	2D24a 연동 설계서	연동 항목 테이블 매핑 설계서
		2D25 자료 구축 설계	2D25a 자료 구축 설계서	
		2D26 시스템 보안 설계	2D26a 시스템 보안 설계서	
	2D3 시스템 설치 준비	2D31 시스템 설치 계획	2D31a 시스템 설치 계획서	

개요

단계	활동	작업	산출물	부록
3T 구현 및 테스트	3T1 구현	3T11 데이터베이스 구축	3T11a 물리적 데이터베이스	
		3T12 컴포넌트 구현	3T12a 컴포넌트 코드	
		3T13 사용자 인터페이스 구현	3T13a 사용자 인터페이스 코드	
	3T2 단위 테스트	3T21 단위 테스트 준비	3T21a 단위 테스트 기술서	
		3T22 단위 테스트 수행		
	3T3 소프트웨어 통합 및 테스트	3T31 소프트웨어 통합 테스트 준비	3T31a 소프트웨어 통합 테스트 기술서	
		3T32 소프트웨어 통합 및 테스트 수행		
	3T4 시스템 통합 및 테스트	3T41 시스템 통합 테스트 준비	3T41a 시스템 통합 테스트 기술서	
		3T42 시스템 통합 및 테스트 수행		
	3T5 지침서 작성	3T51 사용자 지침서 작성	3T51a 사용자 지침서	
		3T52 운용자 지침서 작성	3T52a 운용자 지침서	
4S 인도	4S1 시스템 설치	4S11 시스템 설치 실시	4S11a 시스템 설치 결과서	
	4S2 인수 지원	4S21 인수 지원 실시	-	
4단계	15활동	37작업	35산출물	4부록

■ 신규 버전 전체 프로세스 비교

국방 CBD 방법론 v1.1

활동	작업	산출물
요구 사항 정의 1R1	상위 요구사항 정의(1R11)	상위 요구사항 정의서
	도메인 모델링(1R12)	도메인 명세서 용어집
	현행 시스템 분석(1R13)	현행 시스템 분석서
	비즈니스 모델링(1R14)	비즈니스 정의서
	요구사항 명세(1R15)	요구사항 명세서
아키텍처 정의 1R2	소프트웨어 아키텍처 정의(1R21)	소프트웨어 아키텍처 정의서
	시스템 아키텍처 정의(1R22)	시스템 아키텍처 정의서
	표준 지침 수립(1R23)	표준 지침서
요구 사항 분석 1R3	유스케이스 모델링(1R31)	유스케이스 명세서
	사용자 인터페이스 프로토타입(1R32)	사용자 인터페이스 정의서
	클래스 모델링(1R33)	클래스 명세서
	테스트케이스 정의(1R34)	테스트케이스 정의서
개략 설계 2D1	컴포넌트 식별(2D11)	컴포넌트 목록 컴포넌트 아키텍처 정의서
	컴포넌트 획득 방법 식별(2D12)	컴포넌트 획득 방법 식별서
	인터페이스 상호작용 정의(2D13)	인터페이스 상호작용 명세서
	컴포넌트 명세(2D14)	인터페이스 명세서 컴포넌트 명세서
	사용자 인터페이스 설계(2D15)	사용자 인터페이스 설계서
	데이터 모델링(2D16)	데이터 설계서
	컴포넌트 내부 설계(2D21)	컴포넌트 설계서 트랜잭션 정의서
상세 설계 2D2	컴포넌트 구현 설계(2D22)	컴포넌트 구현 설계서 트랜잭션 정의서(보완)
	사용자 인터페이스 구현 설계(2D23)	사용자 인터페이스 구현 설계서
	데이터베이스 설계(2D24)	데이터베이스 설계서
	테스트 계획(3T11)	테스트 계획서
테스트 준비 3T1	컴포넌트 테스트 설계(3T12)	컴포넌트 테스트 설계서
	데이터베이스 구축(3T21)	물리적 데이터베이스 컴포넌트 코드
구현 3T2	컴포넌트 구현 및 테스트(3T22)	컴포넌트 테스트 결과서
	사용자 인터페이스 구현(3T23)	사용자 인터페이스 코드
통합 테스트 3T3	통합 테스트 설계(3T31)	통합 테스트 설계서
	통합 테스트 수행(3T32)	통합 테스트 결과서
시스템 테스트 3T4	시스템 테스트 설계(3T41)	시스템 테스트 설계서
	시스템 테스트 수행(3T42)	시스템 테스트 결과서
지침서 작성 3T5	사용자 지침서 작성(3T51)	사용자 지침서
	운용자 지침서 작성(3T52)	운용자 지침서
시스템 설치 4S1	시스템 설치 계획(4S11)	시스템 설치 계획서
	시스템 설치 실시(4S12)	시스템 설치 보고서
인수 지원 4S2	인수 테스트 지원(4S21)	-
	사용자 교육(4S22)	교육보고서
총 12 활동	총 37 작업	총 41 산출물

국방 CBD 방법론 v2.0

산출물	작업	활동
1R11a 도메인 정의서	1R11 도메인 모델링	요구사항 식별
1R11b 용어집		
1R12a 현행 시스템 분석서		
1R13a 비즈니스 정의서	1R12 현행 시스템 분석	
	1R13 비즈니스 모델링	
1R14a 요구사항 정의서	1R14 요구사항 정의	1R2 아키텍처 정의
1R21a 시스템 아키텍처 정의서	1R21 시스템 아키텍처 정의	
1R22a 표준 지침서	1R22 표준 지침 수립	
1R31a 유스케이스 정의서	1R31 유스케이스 모델링	1R3 요구사항 분석
1R32a 연동 소요 정의서	1R32 연동 소요 분석	
1R33a 사용자 인터페이스 정의서	1R33 사용자 인터페이스 정의	
1R34a 클래스 정의서	1R34 클래스 모델링	
1R41a 자료 구축 계획서	1R41 자료 구축 계획	1R4 자료 구축 준비
1R51a 테스트 계획서	1R51 테스트 계획	1R5 테스트 준비
2D11a 컴포넌트 아키텍처 명세서	2D11 컴포넌트 식별	2D1 개략 설계
2D12a 인터페이스 상호작용 명세서	2D12 인터페이스 상호작용 명세	
2D13a 컴포넌트 명세서	2D13 컴포넌트 명세	
2D14a 사용자 인터페이스 명세서	2D14 사용자 인터페이스 명세	
2D15a 데이터 명세서	2D15 데이터 모델링	
2D21a 컴포넌트 설계서	2D21 컴포넌트 구현 설계	2D2 상세 설계
2D21b 트랜잭션 설계서		
2D22a 사용자 인터페이스 설계서	2D22 사용자 인터페이스 구현 설계	
2D23a 데이터베이스 설계서	2D23 데이터베이스 설계	
2D24a 연동 설계서	2D24 연동 설계	
2D25a 자료 구축 설계서	2D25 자료 구축 설계	
2D26a 시스템 보안 설계서	2D26 시스템 보안 설계	
2D31a 시스템 설치 계획서	2D31 시스템 설치 계획	2D3 시스템 설치 준비
3T11a 물리적 데이터베이스	3T11 데이터베이스 구축	3T1 구현
3T12a 컴포넌트 코드	3T12 컴포넌트 구현	
3T13a 사용자 인터페이스 코드	3T13 사용자 인터페이스 구현	
3T21a 단위 테스트 기술서	3T21 단위 테스트 준비 3T22 단위 테스트 수행	3T2 단위 테스트
3T31a 소프트웨어 통합 테스트 기술서	3T31 소프트웨어 통합 테스트 준비 3T32 소프트웨어 통합 및 테스트 수행	3T3 소프트웨어 통합 및 테스트
3T41a 시스템 통합 테스트 기술서	3T41 시스템 통합 테스트 준비 3T42 시스템 통합 및 테스트 수행	3T4 시스템 통합 및 테스트
3T51a 사용자 지침서	3T51 사용자 지침서 작성	3T5 지침서 작성
3T52a 운용자 지침서	3T52 운용자 지침서 작성	
4S11a 시스템 설치 결과서	4S11 시스템 설치 실시	
4S21a 인수 지원 실시	4S21 인수 지원 실시	4S2 인수 지원
총 35 산출물	총 37 작업	총 15 활동

산출물 통합 : 산출물명 / 산출물 이동 : 산출물명 / 산출물 삭제 : 산출물명 / 산출물 추가 : 산출물명 / 산출물 내용 변경 : 산출물명

신·구 버전 산출물 서식 비교

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
1R 분석	1R11a 상위 요구사항 정의서	1. 프로젝트 목표 2. 문제 및 기회 분석 3. 상위 요구사항	(삭제)	(타 산출물로 이동)	• 요구사항 명세서와 통합
	1R12a 도메인 명세서	1. 도메인 모델 (1) 개요 (2) 모델 (3) 공통성 정의 2. 시스템 배경 모델 (1) 개요 (2) 모델	1R11a 도메인 정의서	1. 도메인모델 1.1. 개요 1.2. 도메인 구성도 1.3. 공통성 정의	• 시스템 배경 모델을 도메인 모델과 통합 • MND-AF 1.2의 항목 반영
	1R12b 용어집	1. 용어 정의	1R11b 용어집	1. 용어 정의	• MND-AF 1.2와 서식 통일
	1R13a 현행 시스템 분석서	1. 개요 2. 현행 시스템 환경 기술 (1) 하드웨어 (2) 소프트웨어 (3) 네트워크 3. 현행 시스템 구성도 4. 현행 시스템 자료구조 5. 현행 시스템 기능 6. 문제점/해결방안	1R12a 현행 시스템 분석서	1. (현행 시스템명) 1.1. 관련 지침 1.2. 현행 시스템 개요 1.3. 시스템 기능 1.4. 하드웨어 구성 1.4.1. 하드웨어 구성도 1.4.2. 하드웨어 1.5. 네트워크 및 통신장비 구성 1.5.1. 네트워크 구성도 1.5.2. 시스템 노드 1.5.3. 네트워크 1.5.4. 랜 1.5.5. 통신장비 1.6. 소프트웨어 구성 1.6.1. 패키지 소프트웨어 1.6.1. 시스템 소프트웨어 1.7. 데이터베이스 구성 1.7.1. 데이터베이스 1.7.2. 테이블 1.7.3. 코드	• 현행 시스템 분석 항목을 세분화 • 국방정보기술아키텍처의 AS-IS 아키텍처를 활용할 수 있도록 MND-AF 1.2의 항목 반영

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
				1.7.4. 세부코드 1.8. 타 시스템간 인터페이스 1.8.1. 인터페이스 구성도 1.8.2. 인터페이스 1.8.3. 인터페이스 데이터 2. 성능요소 3. 문제점 및 해결방안	
	1R14a 비즈니스 정의서	1. 비즈니스 조직 흐름도 2. 비즈니스 프로세스 흐름도 3. 비즈니스 개념 다이어그램 4. 서브시스템 정의 (1) 서브시스템 목록 (2) 서브시스템 구성도	1R13a 비즈니스 정의서	1. 비즈니스 패키지 1.1. 비즈니스 패키지 구성도 1.2. 비즈니스 패키지 목록 2. 비즈니스 액터 2.1. 비즈니스 액터 목록 3. (비즈니스 패키지명) 3.1. 비즈니스 프로세스 목록 3.2. (비즈니스 프로세스명) 3.2.1. 비즈니스 프로세스도 3.2.2. 비즈니스 규칙 4. 비즈니스 개념 4.1. 비즈니스 개념 구성도 4.2. 비즈니스 개념 목록	<ul style="list-style-type: none"> 패키지 다이어그램 기법 추가 비즈니스 액터, 규칙, 개념 식별 유도 순수 비즈니스 식별을 위해 서브시스템 정의 삭제
	1R15a 요구사항 명세서	1. 요구사항 목록	1R14a 요구사항 정의서	1. 프로젝트 목표 2. 상위 요구사항 3. 요구사항	<ul style="list-style-type: none"> 상위 요구사항 정의서와 통합함에 따라 프로젝트 목표와 상위 요구사항 추가
			(부록) 요구사항 추적표	1. 요구사항 추적표	<ul style="list-style-type: none"> 요구사항을 기준으로 유스 케이스, 컴포넌트 등과의 추적 관계 신규로 정의
	1R21a 소프트웨어 아키텍처 정의서	1. 아키텍처 제약 사항 및 요구사항 2. 초기 소프트웨어 아키텍처 다이어그램 3. 품질 평가 기준	(삭제)	(타 산출물로 이동)	<ul style="list-style-type: none"> 작성 항목 중복으로 시스템 아키텍처 정의서와 통합

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
		4. 소프트웨어 대안 아키텍처 다이어그램 5. 소프트웨어 아키텍처 평가 결과 6. 최종 소프트웨어 아키텍처 다이어그램			
	1R22a 시스템 아키텍처 정의서	1. 비즈니스 패키지 다이어그램 2. 소프트웨어 아키텍처 다이어그램 3. 기반 아키텍처 (1) 기반 아키텍처 다이어그램 (2) H/W 구성 (3) S/W 구성 4. 실행 아키텍처 다이어그램	1R21a 시스템 아키텍처 정의서	1. 개요 1.1. 관련지침 1.2. 개발 대상 시스템 정의 2. 운용 환경 2.1. 개요 2.2. 운용 환경 구성도 3. 하드웨어 구성 3.1. 하드웨어 구성도 3.2. 하드웨어 4. 네트워크 및 통신장비 구성 4.1. 네트워크 구성도 4.2. 시스템 노드 4.3. 네트워크 4.4. 랜 4.5. 통신장비 5. 소프트웨어 구성 5.1. 패키지 소프트웨어 5.2. 시스템 소프트웨어 6. 아키텍처 6.1. 소프트웨어 아키텍처 6.1.1. 소프트웨어 아키텍처 스타일 6.1.1.1. 아키텍처 다이어그램 6.1.1.2. 아키텍처 설명 6.1.2. 아키텍처 구성 요소 6.2. 실행 아키텍처 6.2.1. 실행 아키텍처 스타일 6.2.2. 아키텍처 구성 요소 7. 기술 표준 7.1 국방 표준 7.2 기타 표준	<ul style="list-style-type: none"> • 소프트웨어 아키텍처 정의서와 통합 • 지침, 분산환경, 기술 표준 등 목표 시스템에 영향을 주는 요소의 식별 유도 • 각 작성 항목 상세화로 체계적인 아키텍처 정의 유도 • MND-AF 1.2의 항목 반영

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
	1R23a 표준 지침서	1. 목적 2. 프로젝트 표준 3. 사용자 인터페이스 표준 4. 설계 표준 5. 코딩 표준	1R22a 표준 지침서	1. 프로젝트 표준 2. 분석/설계 표준 3. 사용자 인터페이스 표준 4. 데이터 모델링 표준 5. 개발 표준 6. 기타	<ul style="list-style-type: none"> 표준 작성 항목을 세분화
	1R31a 유스케이스 명세서	1. 시스템 개요 2. 액터 및 유스케이스 정의 (1) 액터 목록 (2) 유스케이스 정의 3. 유스케이스 다이어그램 4. 유스케이스 설명 (1) (유스케이스명) 1) (유발자(initiating actor)명) 2) 개요 3) 사건 흐름 - 기본 흐름 - 대안 흐름 - 예외 흐름 4) 사전/ 사후 조건 5) 가변성 6) 부가사항 5. 유스케이스 시나리오 6. 타 시스템과의 연계 정보 정의	1R31a 유스케이스 정의서	1. 액터 목록 2. (유스케이스 다이어그램명) 2.1. 유스케이스 다이어그램 2.2. (유스케이스명) 2.2.1. 개요 2.2.2. 관련 액터 2.2.3. 사건 흐름 2.2.4. 가변성 2.2.5. 부가사항 2.2.6. 시나리오	<ul style="list-style-type: none"> 시스템 개요는 시스템 아키텍처 정의서로 이동 액터 및 유스케이스 정의 작성 항목은 동일하며 구성 방식 변경 타 시스템과의 연계 정보 정의는 연동 관련 산출물로 이동
	(없음)	(없음)	1R32a 연동 소요 정의서	1. 개요 1.1. 연동 개념도 1.2. 개발 대상 시스템 정의 1.3. 타 시스템 정의 1.4. 연동 대상 시스템 노드 2. 연동 소요 정의 2.1. 인터페이스 정의 3. 연동 환경 분석 3.1. 인터페이스 기술 3.2. 인터페이스 기술 구성 요소	<ul style="list-style-type: none"> 연동 관련 산출물 신규 정의 연동 소요와 제반 환경 분석 유도

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
				3.3. 인터페이스 구성 요소 흐름 3.4. 인터페이스 하드웨어	
			(부록) 연동 항목 소요서(IER)	1. (인터페이스명)	
	1R32a 사용자 인터페이스 정의서	1. 사용자 인터페이스 프로토타입 (1) (화면이름) 1) 화면 2) 화면설명	1R33a 사용자 인터페이스 정의서	1. 메뉴 구성 2. (기능 이름) 2.1. 화면 목록 2.2. 화면 흐름도 2.3. 화면 정의 2.3.1. (화면명(화면ID)) 2.3.1.1 설명 2.3.1.2 레이아웃 2.3.1.3 항목 설명 2.4. 보고서 정의 2.4.1. (보고서명(보고서ID)) 2.4.1.1. 설명 2.4.1.2. 레이아웃 2.4.1.3. 항목 설명	<ul style="list-style-type: none"> • 사용자 인터페이스 프로토타입 작성 대신 전체 화면에 대한 정의를 하는 작업으로 변경 • 분석단계 전체 화면에 대한 사용자 검토 유도 • 보고서 항목 추가
	1R33a 클래스 명세서	1. 클래스 다이어그램 2. 클래스 정의 (1) (클래스명) 1) 개요 2) 속성 3) 오퍼레이션	1R34a 클래스 정의서	1. (클래스 다이어그램명) 1.1. 클래스 다이어그램 1.2. (클래스명) 1.2.1. 개요 1.2.2. 속성 1.2.3. 오퍼레이션	<ul style="list-style-type: none"> • 작성 항목 동일함
	(없음)	(없음)	1R41a 자료 구축 계획서	1. 자료 전환 1.1. (자료 전환 대상명) 1.1.1. 개요 1.1.2. 전환 대상 테이블 1.1.3. 전환 전략 1.1.4. 전환 절차 1.1.5. 전환 일정 1.1.6. 담당 조직 및 세부 임무 1.1.7. 전환 환경	<ul style="list-style-type: none"> • 자료 구축 관련 산출물 신규 정의 • 자료 전환 및 신규 구축 대상을 식별하고 전략 및 절차 정의 유도

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
				2. 초기 자료 구축 2.1. (자료 구축 대상명) 2.1.1. 개요 2.1.2. 구축 전략 2.1.3. 구축 방법 2.1.4. 구축 절차 2.1.5. 구축 일정 2.1.6. 담당 조직 및 세부 임무 2.1.7. 구축 환경	
	1R34a 테스트케이스 정의서	1. 테스트케이스	(삭제)	(삭제)	• 작성 항목 중복으로 각 테스트 기술서와 통합
	3T11a 테스트 계획서	1. 테스트 배경 및 범위 2. 테스트 일정 및 담당자 3. 테스트 유형 및 통과 기준 4. 테스트케이스 선정 기준 5. 테스트 환경 (1) 필요한 소프트웨어 (2) 하드웨어 요구사항	1R51a 테스트 계획서	1. 테스트 종류 및 유형 2. 테스트 특성 및 척도 3. 테스트 수행 계획 4. 테스트 환경	• 작업 시점을 분석 단계로 이동하여 테스트를 고려한 설계 유도 • 작성 항목 정의 명확화
2D 설계	2D11a 컴포넌트 목록	1. 컴포넌트 목록	(삭제)	(타 산출물로 이동)	• 컴포넌트 아키텍처 정의서와 통합
	2D11b 컴포넌트 아키텍처 정의서	1. 컴포넌트 아키텍처 다이어그램 2. 설명	2D11a 컴포넌트 아키텍처 명세서	1. 컴포넌트 목록 2. 컴포넌트 아키텍처 다이어그램 3. 재사용 컴포넌트 활용 방법	• 컴포넌트 목록, 컴포넌트 획득 방법 식별서와 통합
	2D12a 컴포넌트 획득 방법 식별서	1. 컴포넌트 획득 방법 식별	(삭제)	(타 산출물로 이동)	• 컴포넌트 아키텍처 정의서와 통합
	2D13a 인터페이스 상호작용 명세서	1. 인터페이스 상호작용 다이어그램	2D12a 인터페이스 상호작용 명세서	1. (유스케이스명)	• 작성 항목이 동일하며 유스케이스를 기준으로 인터페이스 상호작용 다이어그램 작성 유도

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
	2D14a 인터페이스 명세서	1. 인터페이스 명세 (1) (인터페이스명) 1) 인터페이스 정보 모델	(삭제)	(타 산출물로 이동)	• 컴포넌트 명세서와 통합
	2D14b 컴포넌트 명세서	1. 컴포넌트 명세 (1) (컴포넌트명) - 컴포넌트 다이어그램 - 설명 - 주요 기능 - 인터페이스 - 컴포넌트 상호작용 제약조건 - 인터페이스간 제약조건 - 가변성 - 키워드	2D13a 컴포넌트 명세서	1. (컴포넌트명) 1.1. 컴포넌트 내부 클래스 다이어그램 1.2. 설명 1.3. 인터페이스 1.3.1. (인터페이스명) 1.3.1.1. 오퍼레이션 1.4.1.2. 내부 상호작용 다이어그램 1.4. 내부 클래스 1.4.1 (클래스명) 1.4.1.1. 속성 1.4.1.2. 오퍼레이션 1.4.1.3. 알고리즘 1.5 가변성 1.5.1. (가변성명) 1.5.1.1. 설명 1.5.1.2. 구현 방식	<ul style="list-style-type: none"> • 인터페이스 명세서와 통합 • 주요 기능은 설명과 작성 중복으로 삭제 • 컴포넌트 상호작용 제약조건, 인터페이스간 제약조건은 인터페이스 오퍼레이션의 사전/사후 조건으로 대체 • 내부 상호작용 다이어그램 추가로 상호작용을 통한 오퍼레이션 도출 유도 • 키워드 불필요로 삭제 • 내부 클래스 정의 필요에 따라 컴포넌트 설계서의 내부 클래스 내용의 이동
	2D15a 사용자 인터페이스 설계서	1. 화면 목록 2. 화면 구성 3. 사용자 인터페이스 흐름도 4. 사용자 화면 정의 (1) (화면명(화면 ID)) 1) 설명 2) 화면 레이아웃 3) 화면 항목 설명	2D14a 사용자 인터페이스 명세서	1. 웹 구성 다이어그램 2. 웹 클래스 2.1. 웹 클래스 목록 2.2. (클래스 명) 2.2.1. 속성 2.2.2. 오퍼레이션 2.2.3. 알고리즘 3. 웹 컴포넌트 3.1. 웹 컴포넌트 목록 4. 웹 페이지 4.1. 웹 페이지 목록	<ul style="list-style-type: none"> • 웹 구성 다이어그램, 웹 클래스 설계 작성 항목이 사용자 인터페이스 구현 설계서에서 이동 • 웹 컴포넌트 목록, 웹 페이지 목록 작성 항목 추가 • 화면 정의 작성 항목은 1R33a 사용자 인터페이스 정의서로 이동

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
	2D16a 데이터 설계서	1. 데이터 모형	2D15a 데이터 명세서	1. 논리 데이터 모형	• 작성 항목 동일함
	2D21a 컴포넌트 설계서	1. 컴포넌트 설계 (1) (컴포넌트명) 1) 내부 클래스 다이어그램 2) 내부 클래스 ① (클래스명) - 개요 - 속성 - 오퍼레이션 - 알고리즘 3) 내부 시퀀스 다이어그램 4) 내부 콜래보레이션 다이어그램 5) 가변성	2D21a 컴포넌트 설계서	1. (컴포넌트명) 1.1 구현 클래스 1.1.1 구현 클래스 맵핑 목록 1.1.2 (클래스명) 1.1.2.1 속성 1.1.2.2 오퍼레이션 1.2 인터페이스 1.2.1 (인터페이스명) 1.2.1.1 오퍼레이션 2. 기타 클래스 2.1 (클래스명) 2.1.1 설명 2.1.2 속성 2.1.3 오퍼레이션 2.1.4 알고리즘 3. 패키지 정의 4. 배치 정의 4.1 디플로이먼트 다이어그램 4.2 배포정의	• 작성 항목 중복으로 컴포넌트 구현 설계서와 통합 • 클래스 개요 생략 • 내부 클래스 다이어그램, 내부 클래스 알고리즘은 컴포넌트 명세서로 이동
	2D21b 트랜잭션 정의서	1. 트랜잭션 목록 2. 트랜잭션 명세	2D21b 트랜잭션 설계서	1. 트랜잭션 목록	• 중복 작성 항목 제거 • 예외 처리 정의 유도
	2D22a 컴포넌트 구현 설계서	1. 컴포넌트 구현 명세 (1) (컴포넌트명) 1) 구현 클래스 다이어그램 2) 인터페이스별 오퍼레이션 정의 ① (인터페이스명) 3) 구현 클래스 매핑 2. 기타 클래스 정의 (1) (클래스명) 1) 개요 2) 속성 3) 오퍼레이션 4) 알고리즘	(삭제)	(타 산출물로 이동)	• 컴포넌트 설계서와 통합

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
		3. 패키지 정의 4. 배치 정의 (1) 배포 정의 (2) 디플로이먼트 다이어그램			
	2D23a 사용자 인터페이스 구현 설계서	1. 웹 경로 정의 2. 웹 클래스 설계 (1) 웹 클래스 목록 1) (클래스명) ① 개요 ② 속성 ③ 오퍼레이션 ④ 알고리즘 3. 웹 컴포넌트 목록 4. 웹 페이지 목록 5. 웹 구성 다이어그램	2D22a 사용자 인터페이스 설계서	1. 웹 경로 목록 1.1 웹 클래스 경로 할당 1.2 (웹 구성 요소명) 경로 할당	<ul style="list-style-type: none"> 웹 구성 다이어그램, 웹 클래스 설계 작성 항목은 2D14a 사용자 인터페이스 명세서로 이동
	2D24a 데이터베이스 설계서	1. 테이블 정의 2. 뷰(view) 정의 (1) 뷰 구조 (2) 뷰 정의 3. 인덱스 정의 4. 데이터베이스 정의 5. 파일 구조	2D23a 데이터베이스 설계서	1. 데이터베이스 1.1. DBMS 1.2. 데이터베이스 공간 1.3. 사용자 2. 물리 데이터 모델 2.1. 물리 데이터 모형 2.2. 테이블 및 컬럼 2.2.1. (테이블ID) 2.3. 인덱스 2.3.1. (인덱스ID) 2.4. 시퀀스 3. 데이터베이스 프로그램 3.1. 뷰 3.1.1. 뷰 구조 3.1.2. (뷰ID) 3.2. 패키지 3.2.1. (패키지ID) 3.3. 프로시저 3.3.1. (프로시저ID)	<ul style="list-style-type: none"> 데이터베이스 정의부터 Top-down 방식으로 정의 사용자 정의 및 각 요소의 사용 권한 정의 추가 물리 데이터 모형 추가 시퀀스, 패키지, 프로시저, 함수, 트리거, 스케줄 등 사용 빈도가 높은 요소의 정의 추가

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
				3.4. 함수 3.4.1. (함수ID) 3.5. 트리거 3.5.1. (트리거ID) 3.6. 스케줄 4. 파일 구조	
			(부록) 코드 설계서	1. 코드 목록 2. 세부 코드	• 코드 관련 산출물 신규 정의
	(없음)	(없음)	2D24a 연동 설계서	1. 연동 프로그램 1.1. (인터페이스명) 2. 공통 정보 정의 2.1. 연동 메타 정보 테이블 정의 2.1.1. (인터페이스명) 2.1.1.1. (테이블ID) 2.2. 연동 관련 코드 정의 2.2.1. (인터페이스명) 2.3. 오류 메시지 정의 2.3.1. (인터페이스명) 2.4. 코드 자료 매핑 정의 2.4.1. (인터페이스명) 2.5. 연동 테이블 공통 컬럼 정의 2.5.1. (인터페이스명) 2.6. 연동 메시지 형식 정의 2.6.1 (인터페이스명) 2.6.1.1. (메시지명) 3. 인터페이스 데이터 목록 4. 인터페이스 데이터 상세 4.1. (연동 항목ID)	<ul style="list-style-type: none"> • 연동 관련 산출물 신규 정의 • 코드 식별 및 매핑 방법 정의 유도 • 공통 정보 정의 유도 • 연동 요소의 물리적 설계 유도
			(부록) 연동 항목 테이블 매핑 설계서	1. (연동 항목 ID)	<ul style="list-style-type: none"> • 연동 관련 산출물 신규 정의 • 연동 항목과 테이블 자료의 매핑 로직 설계 유도
	(없음)	(없음)	2D25a 자료 구축 설계서	1. 전환 코드 매핑 2. 전환 자료 매핑	• 자료 구축 관련 산출물 신규 정의

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
				2.1 (자료 전환 대상명) 3. 초기 구축 자료 매핑 3.1 (구축 자료 대상명)	<ul style="list-style-type: none"> 코드 식별 및 매핑 방법 정의 유도 전환 및 구축 자료 매핑 로직 설계 유도
	(없음)	(없음)	2D26a 시스템 보안 설계서	1. 관련 지침 2. 시스템 보안 설계 2.1. 보안정책 구조에 따른 보안 설계 2.2. 기타 보안 설계	<ul style="list-style-type: none"> 시스템 보안 관련 산출물 신규 정의 MND-AF 1.2의 항목 반영
	4S11a 시스템 설치 계획서	1. 시스템 설치 계획	2D31a 시스템 설치 계획서	1. 설치 대상 목록 2. 설치 작업 내용 2.1. 네트워크 통신장비 2.1.1. (이름) 2.2. 하드웨어 2.2.1. (이름) 2.3. 패키지/시스템 소프트웨어 2.3.1. (이름) 2.4. 응용 소프트웨어 2.4.1. (이름) 2.5. 구축 데이터 2.5.1. (이름)	<ul style="list-style-type: none"> 상용 제품 도입 작업의 특성에 따라 작업 시점을 설계 단계로 이동, 설치 제반 사항에 대한 사전 검토를 유도 설치 요소 세분화로 누락 없는 계획 수립 유도 설치 절차 정의와 위험 대책 수립을 유도
3T 구현 및 테스트	3T11a 테스트 계획서	1. 테스트 배경 및 범위 2. 테스트 일정 및 담당자 3. 테스트 유형 및 통과 기준 4. 테스트케이스 선정 기준 5. 테스트 환경 (1) 필요한 소프트웨어 (2) 하드웨어 요구사항	(분석 단계로 이동)	(분석 단계로 이동)	
	3T12a 컴포넌트 테스트 설계서	1. 테스트 대상 및 범위 2. 테스트케이스 설계 (1) (컴포넌트명)	3T21a 단위 테스트 기술서	1. (단위) 테스트 1.1. 테스트 설계 1.1.1. (단위명)	<ul style="list-style-type: none"> 명칭 변경으로 컴포넌트가 아닌 소프트웨어에 대한

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
		1) 테스트케이스 2) 테스트케이스 수행 절차		1.1.1.1. 테스트케이스 1.1.1.2. 테스트 수행 절차	테스트도 정의 유도 • 테스트 대상 및 범위 생략
	3T21a 물리적 데이터베이스	(없음)	3T11a 물리적 데이터베이스	(없음)	
	3T22a 컴포넌트 코드	(없음)	3T12a 컴포넌트 코드	(없음)	
	3T22b 컴포넌트 테스트 결과서	1. 컴포넌트 테스트 결과 (1) (컴포넌트명) 1) 테스트 결과 2) 오류 수정 3) 평가	3T21a 단위 테스트 기술서	1.2 테스트 결과 1.1.2 (단위명) 1.3 평 가	• 산출물 통합으로 중복 작성 요인 제거 • 평가 대상 및 방법 명확화
	3T23a 사용자 인터페이스 코드	(없음)	3T13a 사용자 인터페이스 코드	(없음)	
	3T31a 통합 테스트 설계서	1. 테스트 대상 및 범위 2. 테스트케이스 설계 3. 테스트케이스 수행 절차	3T31a 소프트웨어 통합 테스트 기술서	1. 테스트 설계 1.1. 테스트케이스 1.2. 테스트 수행 절차 2. 테스트 결과 3. 평 가	• 명칭 변경으로 테스트 대상 및 범위 명확화 • 산출물 통합으로 중복 작성 요인 제거 • 평가 대상 및 방법 명확화 • 승인 생략
	3T32a 통합 테스트 결과서	1. 테스트 결과 2. 오류 수정 3. 평가 4. 승인			
	3T41a 시스템 테스트 설계서	1. 테스트 대상 및 범위 2. 테스트케이스 설계 3. 테스트케이스 수행 절차	3T41a 시스템 통합 테스트 기술서	1. 테스트 설계 1.1. 테스트케이스 1.2. 테스트 수행 절차 2. 테스트 결과 3. 평 가	• 명칭 변경으로 테스트 대상 및 범위 명확화 • 산출물 통합으로 중복 작성 요인 제거 • 평가 대상 및 방법 명확화 • 승인 생략
	3T42a 시스템 테스트 결과서	1. 테스트 결과 2. 오류 수정 3. 평가 4. 승인			
	3T51a 사용자 지침서	1. 개요 2. 사용 지침 (1) 설치 방법 (2) 시작 및 종료 방법	3T51a 사용자 지침서	1. 개요 2. 사용 지침 2.1. 설치 방법 2.2. 시작 및 종료 방법	• 작성 항목 동일함 • 비즈니스 흐름 안내 유도 • 중복 작성 요소 제거

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
		(3) 공통 사항 3. 기능별 사용법 (1) (기능 이름) 1) 기능 설명 2) 주의 사항 3) 오류 메시지 및 원인 A. 부록 (1) 오류 메시지 및 처리 절차 (2) 용어 (3) 색인		2.3. 공통 사용 지침 3. 기능별 사용 방법 3.1. (기능명) 3.1.1. (화면명(화면ID)) 3.1.1.1. 비즈니스 흐름 3.1.1.2. 사용 방법 3.1.1.3. 주의 사항 3.1.1.4. 오류 메시지 및 처리 방법	<ul style="list-style-type: none"> 용어 및 색인은 용어집으로 통합
	3T52a 운용자 지침서	1. 개요 2. 시스템 설치 (1) 시스템 구성 1) 시스템 구성도 2) 하드웨어 구성 3) 소프트웨어 구성 4) 네트워크 구성 (2) 시스템 설치 절차 3. 시스템 운용 (1) 자료 관리 (2) 백업 관리 (3) 장애 내역 및 복구 절차 4. 사용 지침 (1) 시작 및 종료 방법 (2) 공통 사항 5. 기능별 사용법 (1) (기능명) 1) 기능 설명 2) 주의 사항 3) 오류 메시지 및 해결 방법 A. 부록 (1) 오류 메시지 및 처리 절차 (2) 용어 (3) 색인	3T52a 운용자 지침서	1. 개요 2. 시스템 구성 2.1. 시스템 구성도 2.2. 운용 환경 구성 2.3. 하드웨어 구성 2.4. 소프트웨어 구성 2.5. 네트워크 구성 3. 시스템 설치 절차 3.1. 네트워크 통신장비 3.2. 하드웨어 3.3. 패키지/시스템 소프트웨어 3.4. 응용 소프트웨어 4. 시스템 운용 4.1. 시작 및 종료 4.2. 자료 백업 4.3. 정기 작업 4.4. 장애 복구 4.5. 보안 유의사항 5. 기타사항 부록.	<ul style="list-style-type: none"> 분산 환경에 대한 안내 유도 시스템 설치 절차 세분화로 누락 없는 안내 유도 정기 예방 작업 정의로써 예방 정비 안내 유도 보안 유의사항 정의 유도 기능별 사용법은 사용자 지침서에서 작성토록 하여 일관성 확보 관련 연락처 등 기재 유도 중복 작성 요소 제거 용어 및 색인은 용어집으로 통합

단계	v1.1		v2.0		비고
	산출물명	작성 항목	산출물명	작성 항목	
4S 인도	4S11a 시스템 설치 계획서	1. 시스템 설치 계획	(설계 단계로 이동)	(설계 단계로 이동)	
	4S12a 시스템 설치보고서	1. 개요 2. 플랫폼 설치 3. 데이터 전환 4. 응용 시스템 설치	4S11a 시스템 설치 결과서	1. 설치 작업 결과 1.1. 네트워크 통신장비 1.1.1 (이름) 1.2. 하드웨어 1.2.1 (이름) 1.3. 패키지/시스템 소프트웨어 1.3.1 (이름) 1.4. 응용 소프트웨어 1.4.1 (이름) 1.5. 구축 데이터 1.5.1 (이름)	<ul style="list-style-type: none"> 설치 대상의 명확화
	4S22a 교육보고서	1. 사용자 교육	(삭제)	(삭제)	<ul style="list-style-type: none"> 관리 프로세스에 해당하는 산출물은 정의하지 않음

■ 신규 버전 지침서 구성 비교

v1.1 지침서 구성		v2.0 지침서 구성		비고
구분	목차	구분	목차	
본문	1. 개요 2. 세부절차	1권	1. 개요 2. 세부절차 3. 테일러링 지침 4. 절차 및 서식 설명 5. 국방정보기술아키텍처 활용 가이드	<ul style="list-style-type: none"> 테일러링의 중요성에 따라 테일러링 지침을 1권에 포함 부록1의 설명서와 산출물 양식을 절차 및 서식 설명으로 통합 v1.1에서 단계, 활동, 작업으로 구성되었던 설명서를 v2.0에서는 단계에 따른 전체 공정을 설명하는 단계 설명서와 해당 작업의 의미와 수행 방법을 제시하는 작업 설명서로 구분 시스템 개발 시 국방정보기술아키텍처 구축 결과를 참고하거나 단위체계 아키텍처 구축에 도움이 되도록 국방정보기술아키텍처 활용 가이드 추가
부록1	1. 설명서 2. 산출물 양식	2권	산출물 예시	<ul style="list-style-type: none"> v1.1에서는 모범 사례로 정의하였으나 v2.0에서는 예시로 정의 v1.1의 모범 사례에서는 소규모 응용 프로그램을 대상으로 하였으나, v2.0에서는 대규모 통합 정보시스템 수준의 예시를 제시
부록2	1. 테일러링 지침 2. 기법서		기법서	<ul style="list-style-type: none"> v1.1 지침서에서 제시된 19개 기법의 내용을 보완 14개 기법을 신규로 추가
부록3	1. 재사용 자산 규격화 2. 용어정의		용어 정의	<ul style="list-style-type: none"> 재사용 자산 규격화는 개발 방법론의 범위를 해당되지 않으므로 제외 필요한 용어에 대해 추가 정의

Procedures

세부 절차

세부 절차

개 요

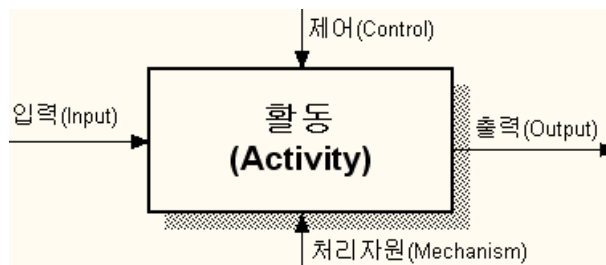
세부 절차에서는 국방 CBD 방법론의 절차를 IDEF0(Integration DEfinition for function modeling) 표기법을 사용하여 상세히 설명하고 있다.

IDEF0는 기업 및 조직의 실체를 추상화하여 모델링하고, 작성된 모델의 체계적인 분석을 통해서 개선된 기업 모델을 설계할 수 있도록 개발된 시스템 분석 및 설계 언어이다. 이는 조직이나 시스템의 의사결정, 행동, 활동을 모델링하고 시스템 개발과 관련된 사람들 간의 의사소통을 촉진하기 위한 언어로 개발되었다.

IDEF0는 기능의 분할(Function Decomposition)과 기능 간의 관계를 유형화함으로써 시스템의 기능(What)을 분석하기 위한 목적으로 개발되었고, 분석하고자 하는 업무의 범위(Scope), 목적(Purpose), 관점(Viewpoint)을 초기에 명확하게 정의하여 하향식(Top-Down)으로 모델링하는 방법이다.

IDEF0의 특징은 업무를 계층구조로 나누어 볼 수 있으며, 각 활동에 비용, 시간, 인원 등의 자원 요소를 적용하여 관점별 비용 산출을 지원할 수 있고, 자료와 업무를 분리하여 고려함으로써 분석의 복잡함을 피할 수 있다.

읽는 법



가. 활동(Activity)

활동은 결과를 산출해 내는 이름이 붙여진 처리, 기능, 직무이다. 활동은 그래픽 다이어그램에서 동사나 동사구로 쓰인 사각형 박스의 형태로 표현된다.

나. ICOM(Input, Control, Output, Mechanism)

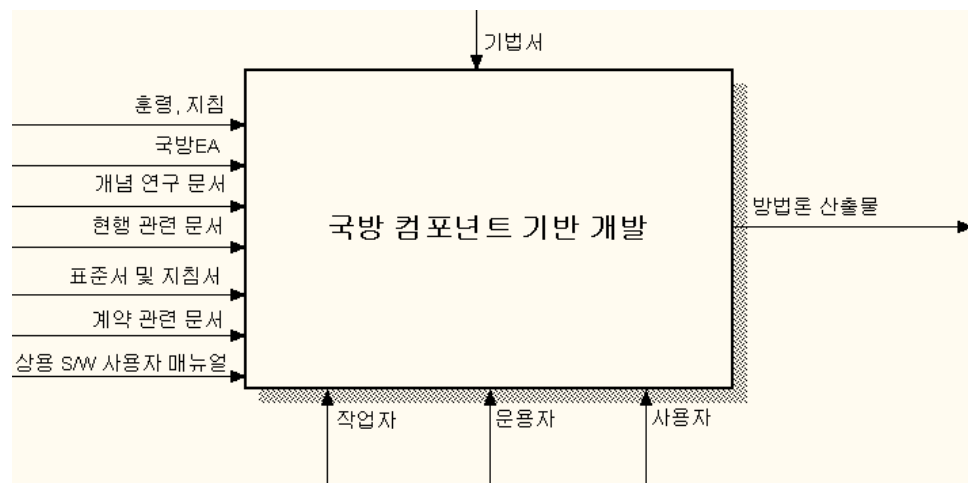
- 입력(Input) : 활동이 출력을 생산하기 위해 사용하는 정보 또는 자료로서 처리 대상이다.
- 제어(Control) : 활동을 통제하는 정보 및 자료로서 처리 대상이 아니다. 제어는 처리 일관성을 보장하기 위해 입력이 출력으로 변환하는 것을 통제한다. (예 : 규정, 표준, 방침, 처리 규칙, 기법 등)

세부 절차

- 출력(Output) : 활동에 의해 생산된 산출물 또는 정보, 자료이다.
- 처리 자원(Mechanism) : 활동을 수행하거나 도움을 주는 사람, 기계, 또는 기존의 체계와 같은 자원이다. 처리 자원은 입력처럼 보일 수 있으나, 입력은 어떤 형태로든 출력의 일부분이 되는데 비해, 처리 자원은 사용은 되나 처리 대상은 아니다.

다. 배경도(Context Diagram)

분석하고자 하는 업무의 최상위 다이어그램으로 해당 업무와 외부와의 인터페이스를 표현한다. 이것은 모델화되고 있는 전체의 주제를 나타낸다.



라. 분해도(Decomposition Diagram)

분해도는 복잡한 활동을 보다 작고, 단순하고, 상세한 활동으로 표현한 그림이다. 분해도 상의 자식(child) 노드는 부모(parent) 노드의 내부를 보여준다. 분해도는 한 부모 노드의 모든 자식 노드를 포함한다.

세부 절차

■ 국방 컴포넌트 기반 개발 방법론 활동 모델

국방 컴포넌트 기반 개발

1R 분석

- 1R1 요구 사항 식별
- 1R2 아키텍처 정의
- 1R3 요구사항 분석
- 1R4 자료 구축 준비
- 1R5 테스트 준비

2D 설계

- 2D1 개략 설계
- 2D2 상세 설계
- 2D3 시스템 설치 준비

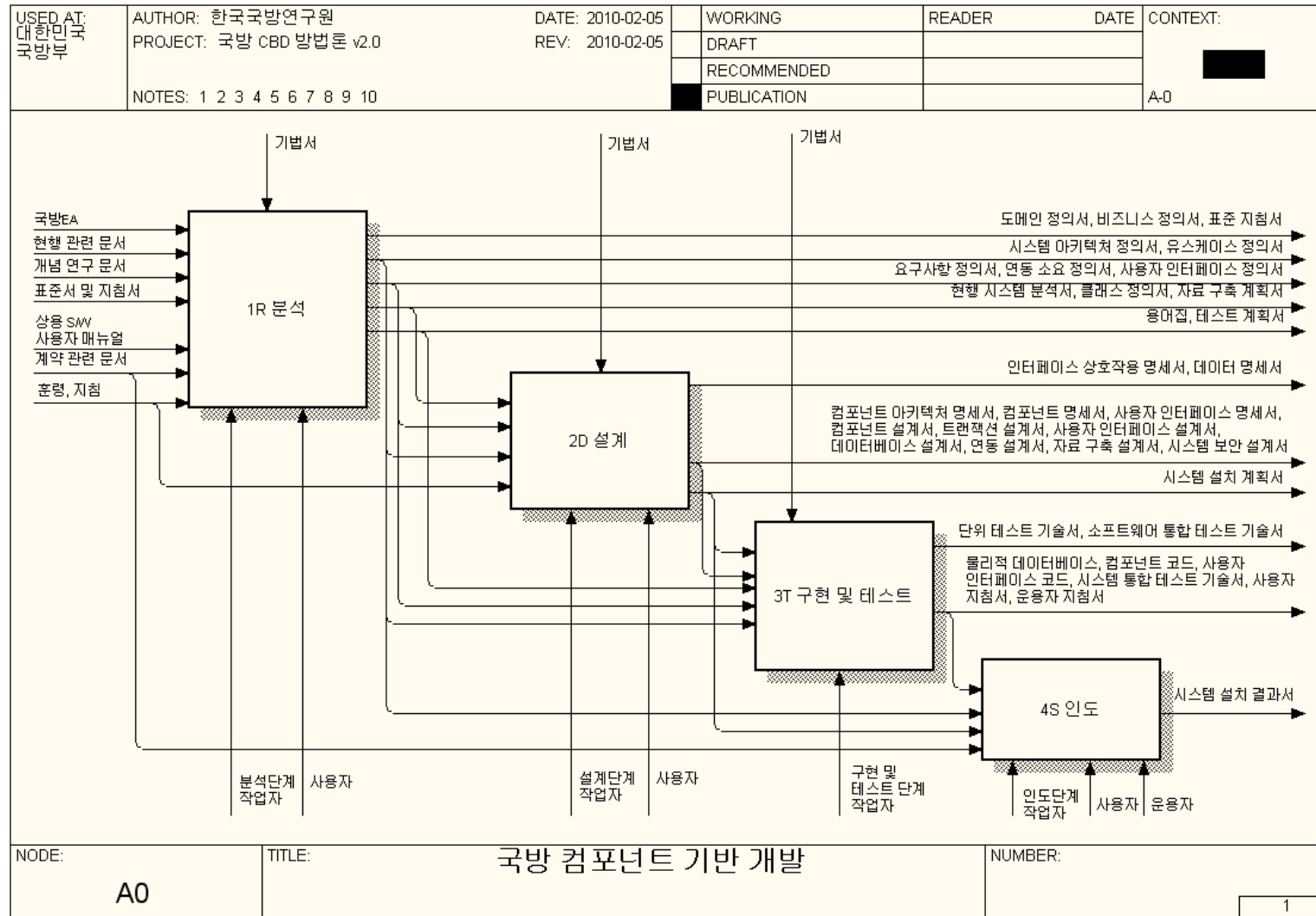
3T 구현 및 테스트

- 3T1 구현
- 3T2 단위 테스트
- 3T3 소프트웨어 통합 및 테스트
- 3T4 시스템 통합 및 테스트
- 3T5 지침서 작성

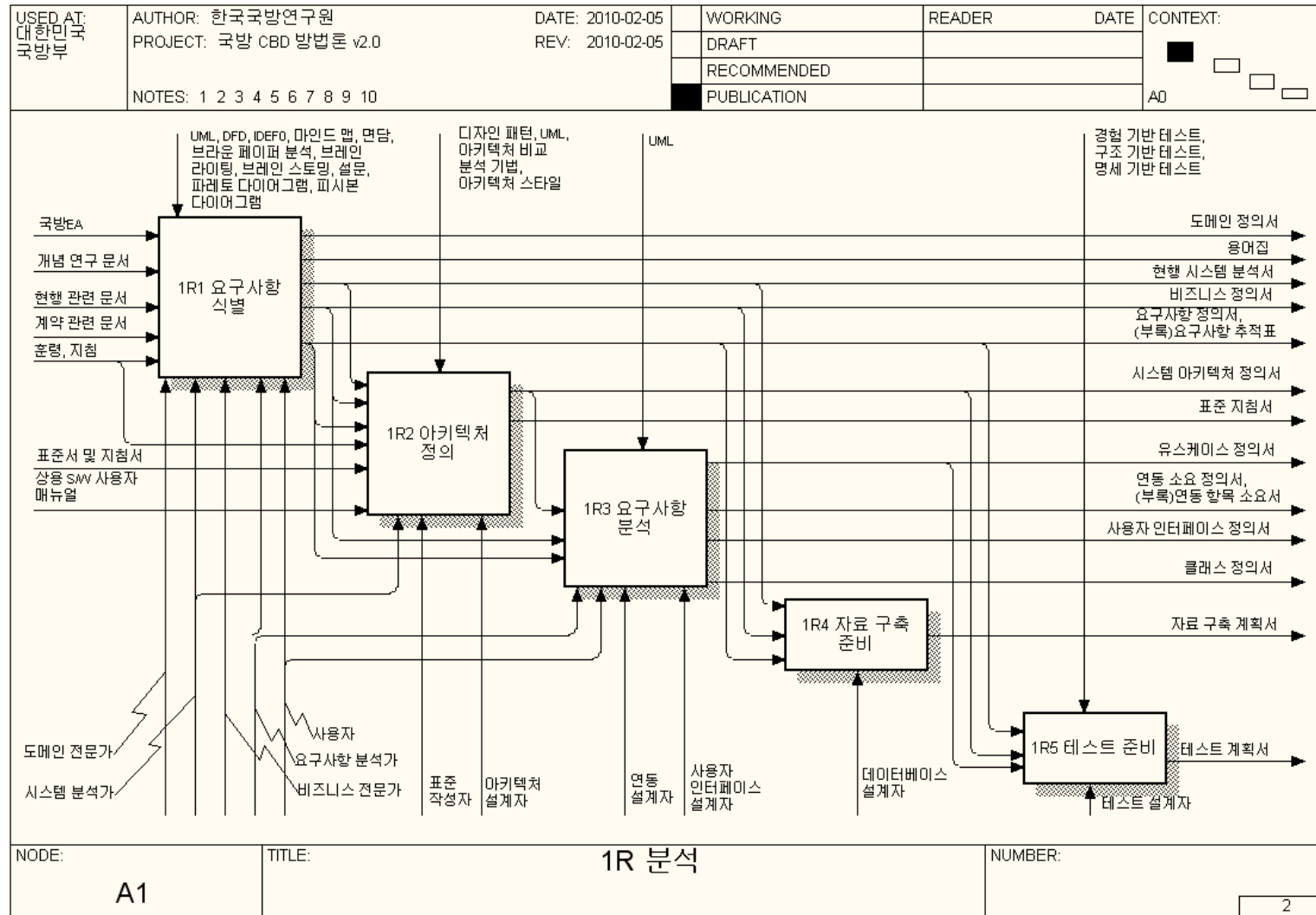
4S 인도

- 4S1 시스템 설치
- 4S2 인수 지원

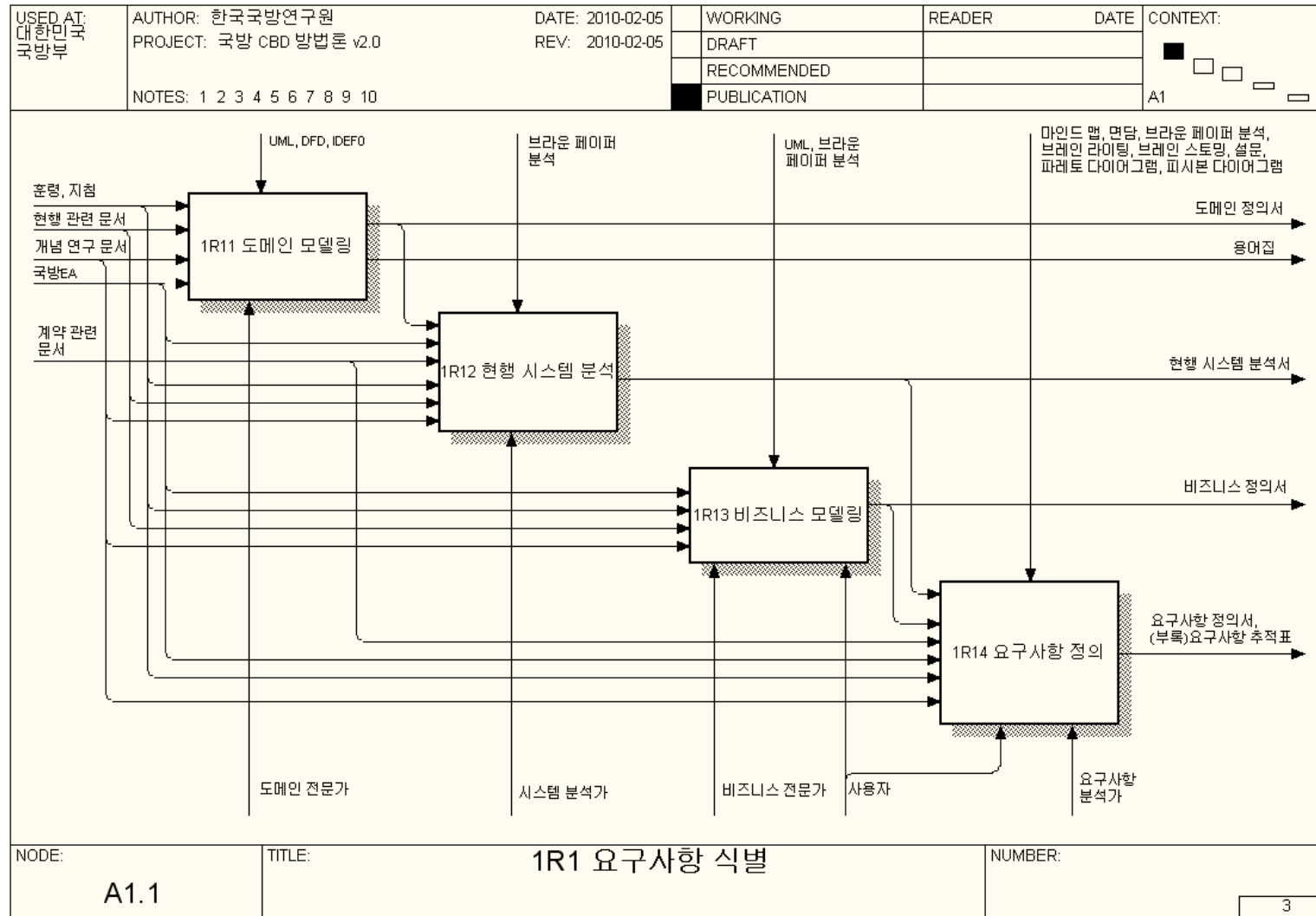
세부 절차



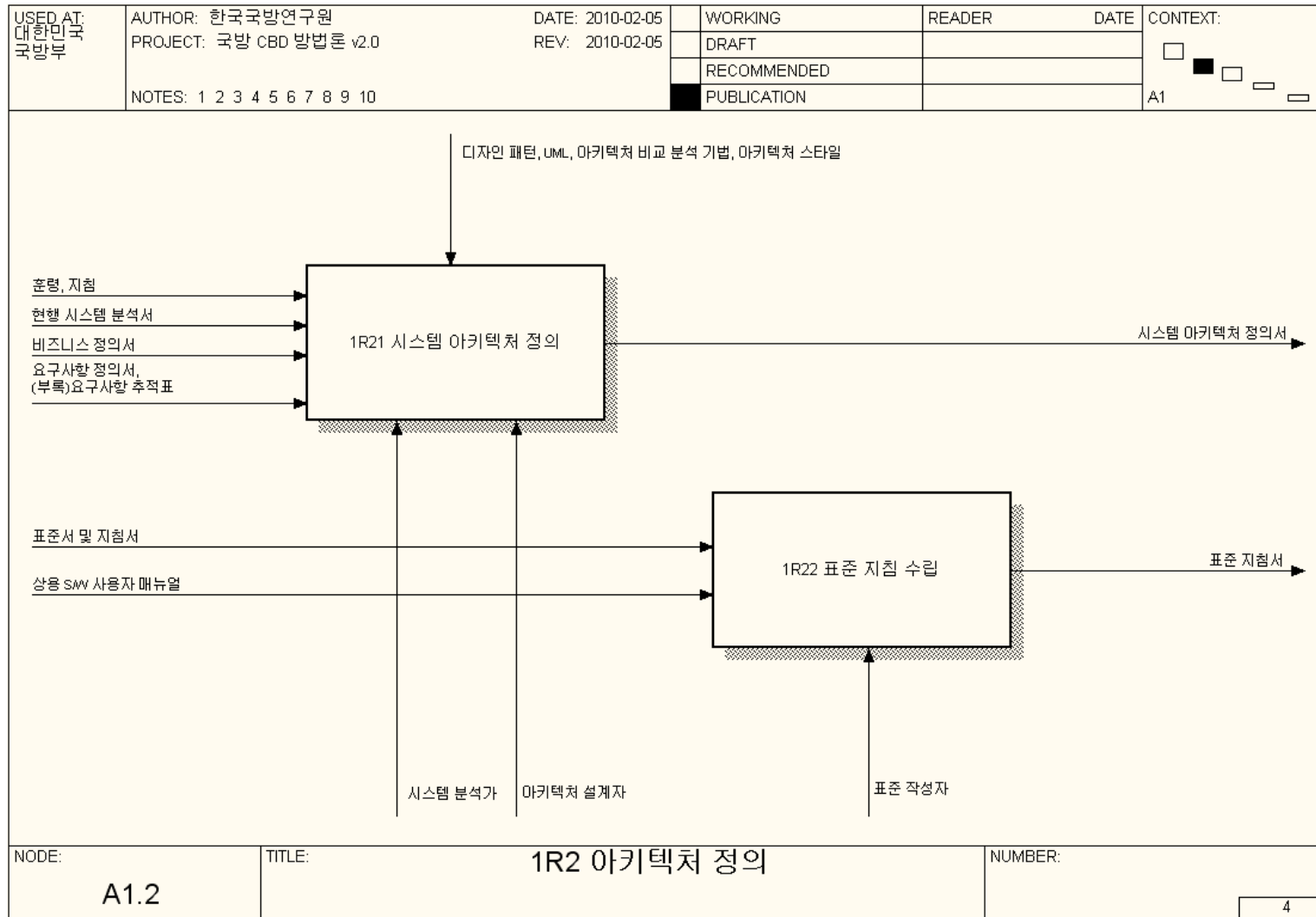
세부 절차



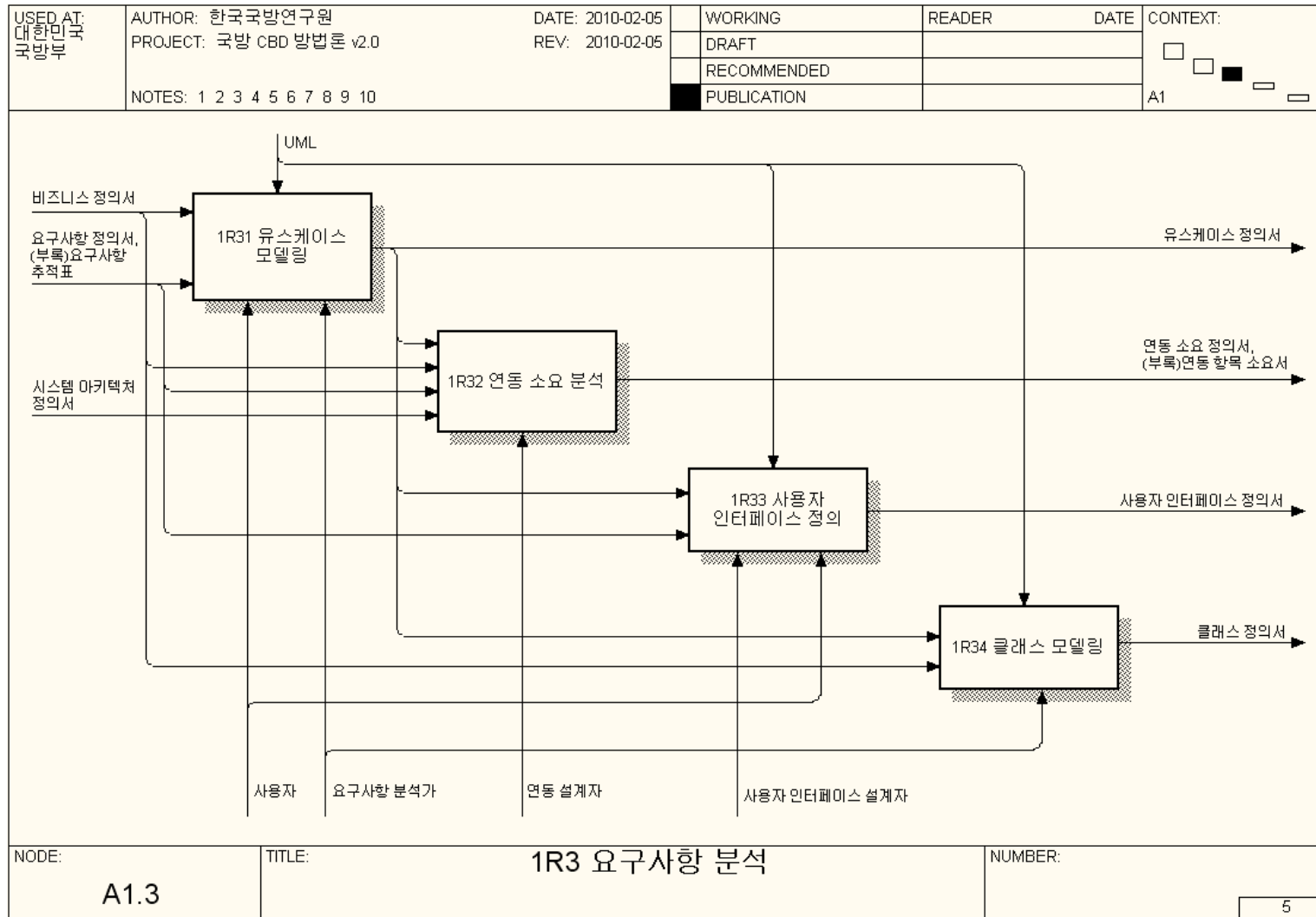
세부 절차



세부 절차



세부 절차



세부 절차

USED AT: 대한민국 국방부	AUTHOR: 한국국방연구원	DATE: 2010-02-05	WORKING	READER	DATE	CONTEXT: □ □ □ □ A1
	PROJECT: 국방 CBD 방법론 v2.0	REV: 2010-02-05	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						

현행 시스템 분석서

비즈니스 정의서

요구사항 정의서,
(부록)요구사항 추적표

1R41 자료 구축 계획

데이터베이스 설계자

자료 구축 계획서

NODE: A1.4	TITLE: 1R4 자료 구축 준비	NUMBER: <div style="border: 1px solid black; width: 50px; text-align: center;">6</div>
----------------------	-------------------------------	---

세부 절차

USED AT: 대한민국 국방부	AUTHOR: 한국국방연구원	DATE: 2009-12-09	WORKING	READER	DATE	CONTEXT: □ □ □ □ □ A1
	PROJECT: 국방 CBD 방법론 v2.0	REV: 2010-03-04	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						

경험 기반 테스트,
구조 기반 테스트,
명세 기반 테스트

↓

요구사항 정의서, (부록)요구사항
추적표

→

시스템 아키텍처 정의서

→

유스케이스 정의서

→

1R51 테스트 계획

→

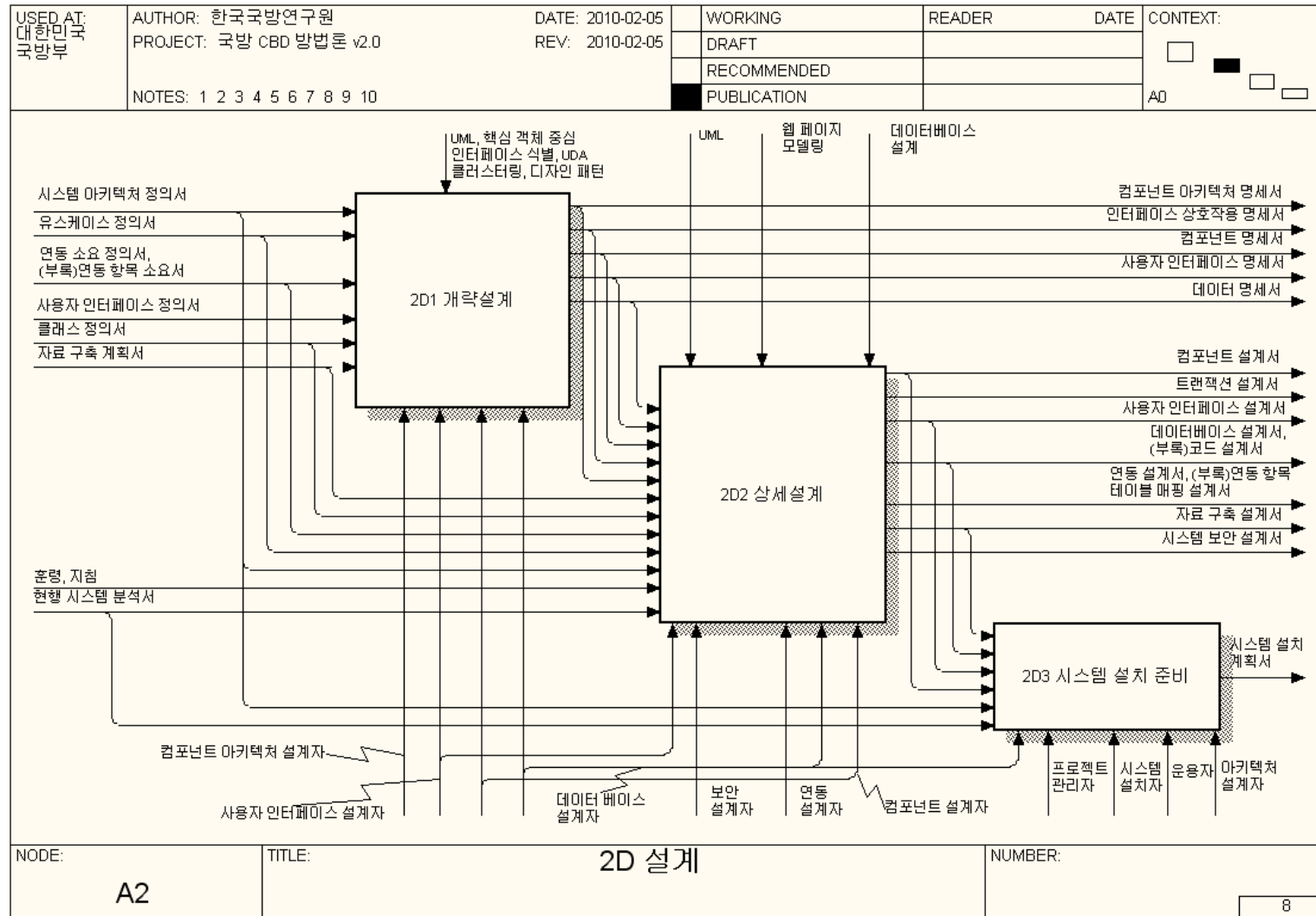
테스트 계획서

↑

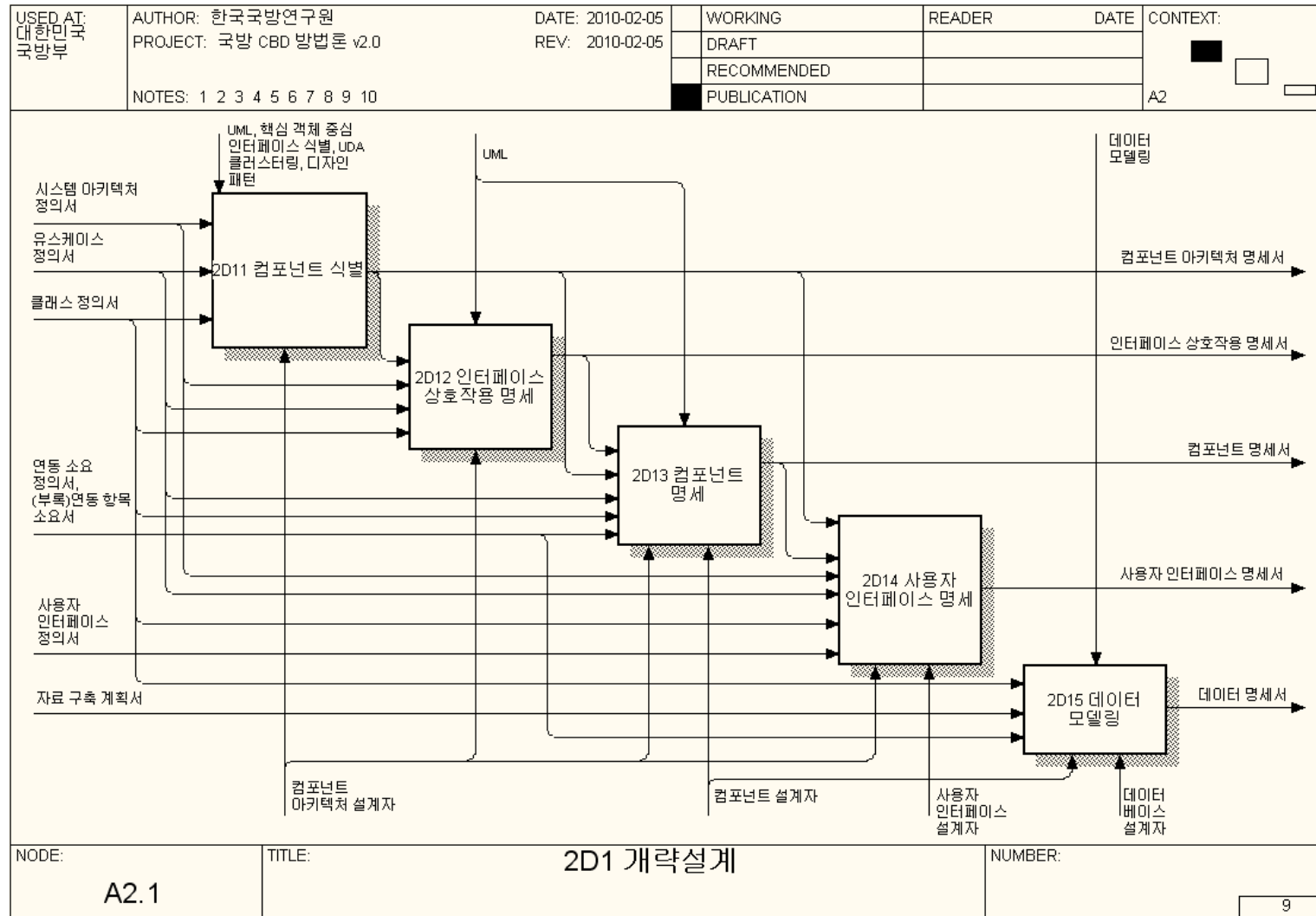
테스트 설계자

NODE: A1.5	TITLE: 1R5 테스트 준비	NUMBER: <div style="border: 1px solid black; width: 50px; text-align: center;">7</div>
--------------------------	---------------------------------	---

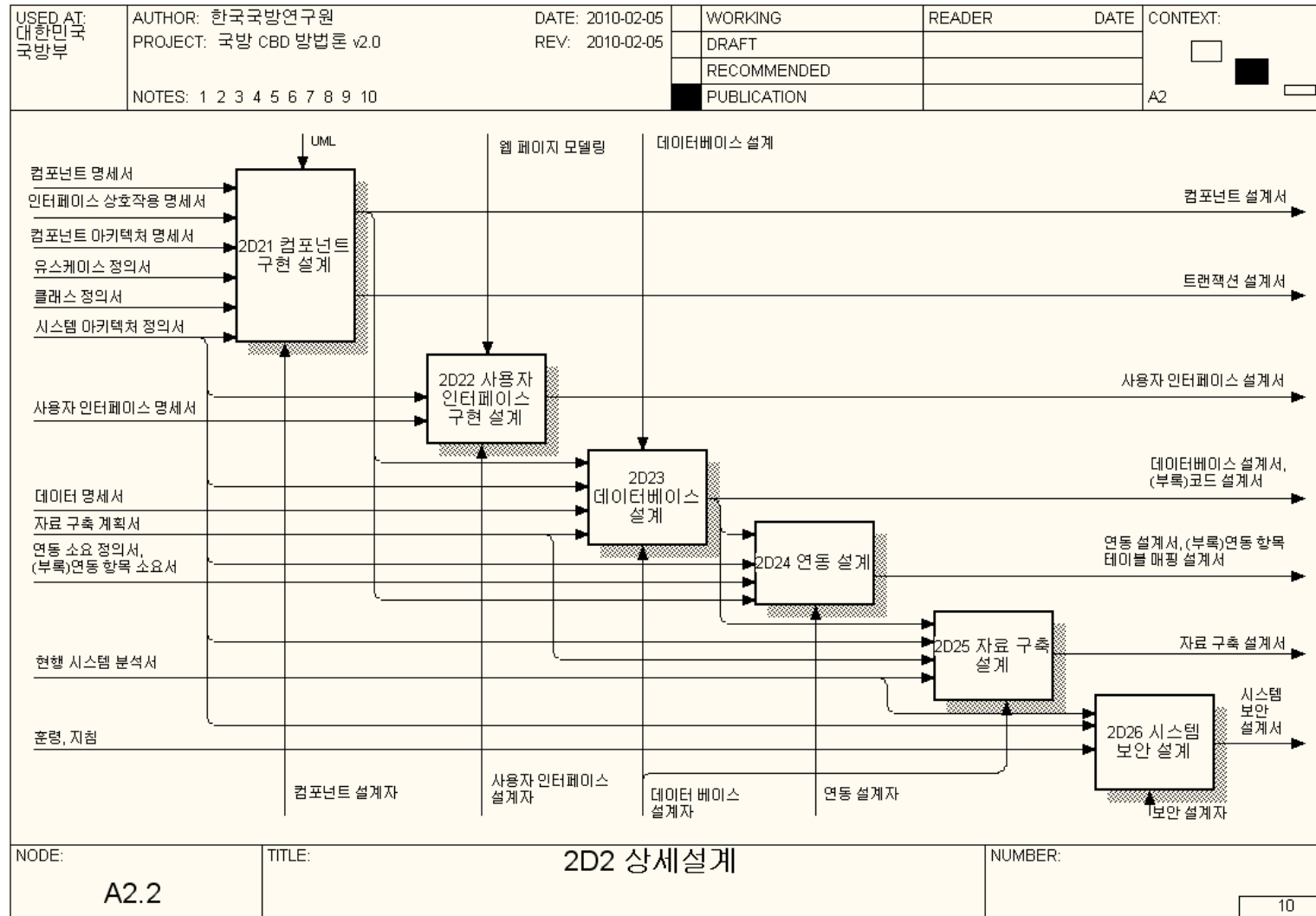
세부 절차



세부 절차



세부 절차



세부 절차

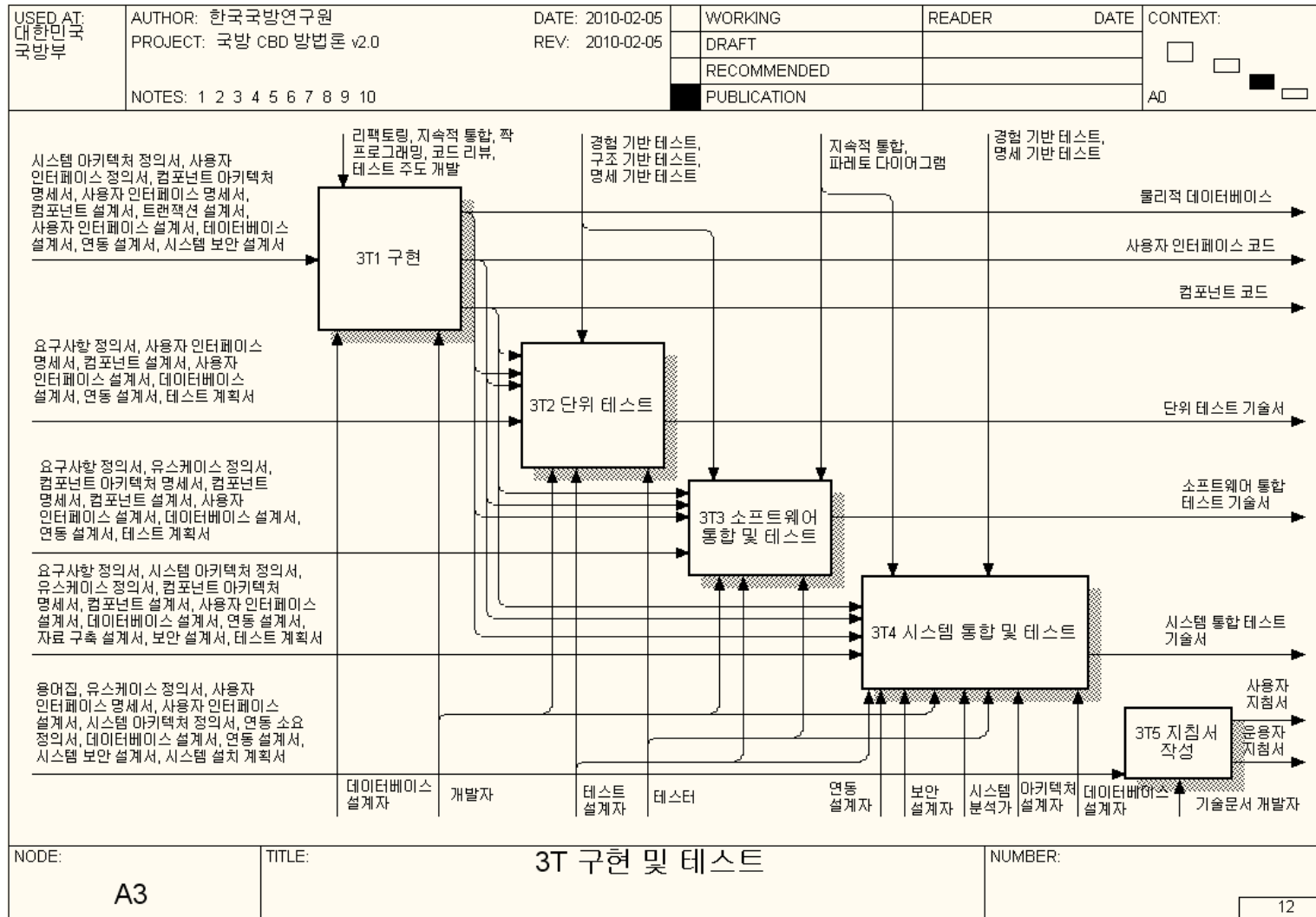
USED AT: 대한민국 국방부	AUTHOR: 한국국방연구원 PROJECT: 국방 CBD 방법론 v2.0	DATE: 2009-12-11 REV: 2010-03-04	WORKING	READER	DATE	CONTEXT: <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> A2
			DRAFT			
			RECOMMENDED			
	NOTES: 1 2 3 4 5 6 7 8 9 10		PUBLICATION			


```

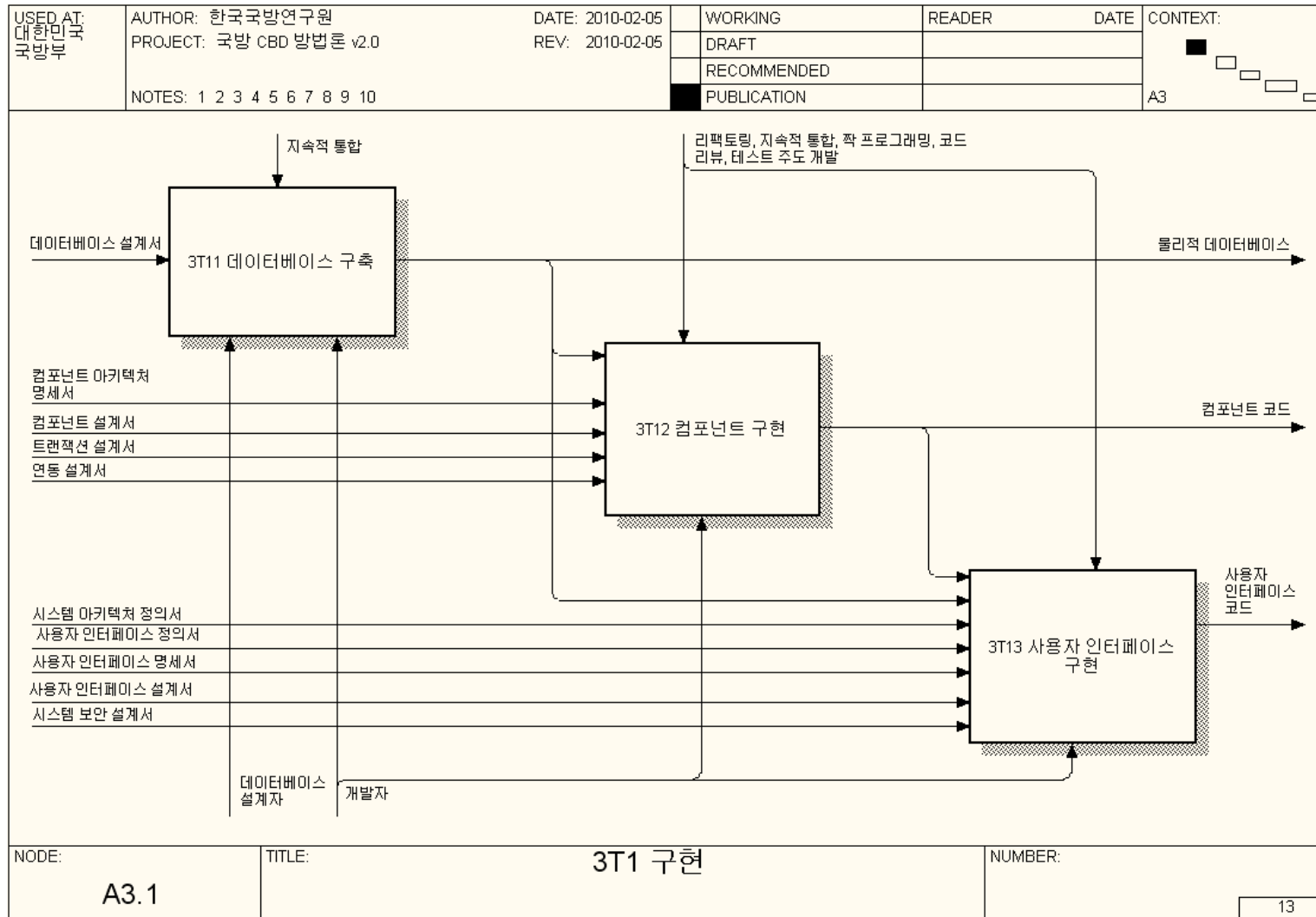
graph LR
    A[시스템 아키텍처 정의서] --> E[2D3 시스템 설치 계획]
    B[현행 시스템 분석서] --> E
    C[데이터베이스 설계서, (부록)코드 설계서] --> E
    D[사용자 인터페이스 설계서] --> E
    F[컴포넌트 설계서] --> E
    G[자료 구축 설계서] --> E
    H[데이터베이스 설계자] --> E
    I[프로젝트 관리자] --> E
    J[시스템 설치자] --> E
    K[운영자] --> E
    L[아키텍처 설계자] --> E
    E --> M[시스템 설치 계획서]
      
```

NODE: A2.3	TITLE: 2D3 시스템 설치 준비	NUMBER: <div style="border: 1px solid black; width: 50px; text-align: center; margin: 0 auto;">11</div>
----------------------	--------------------------------	--

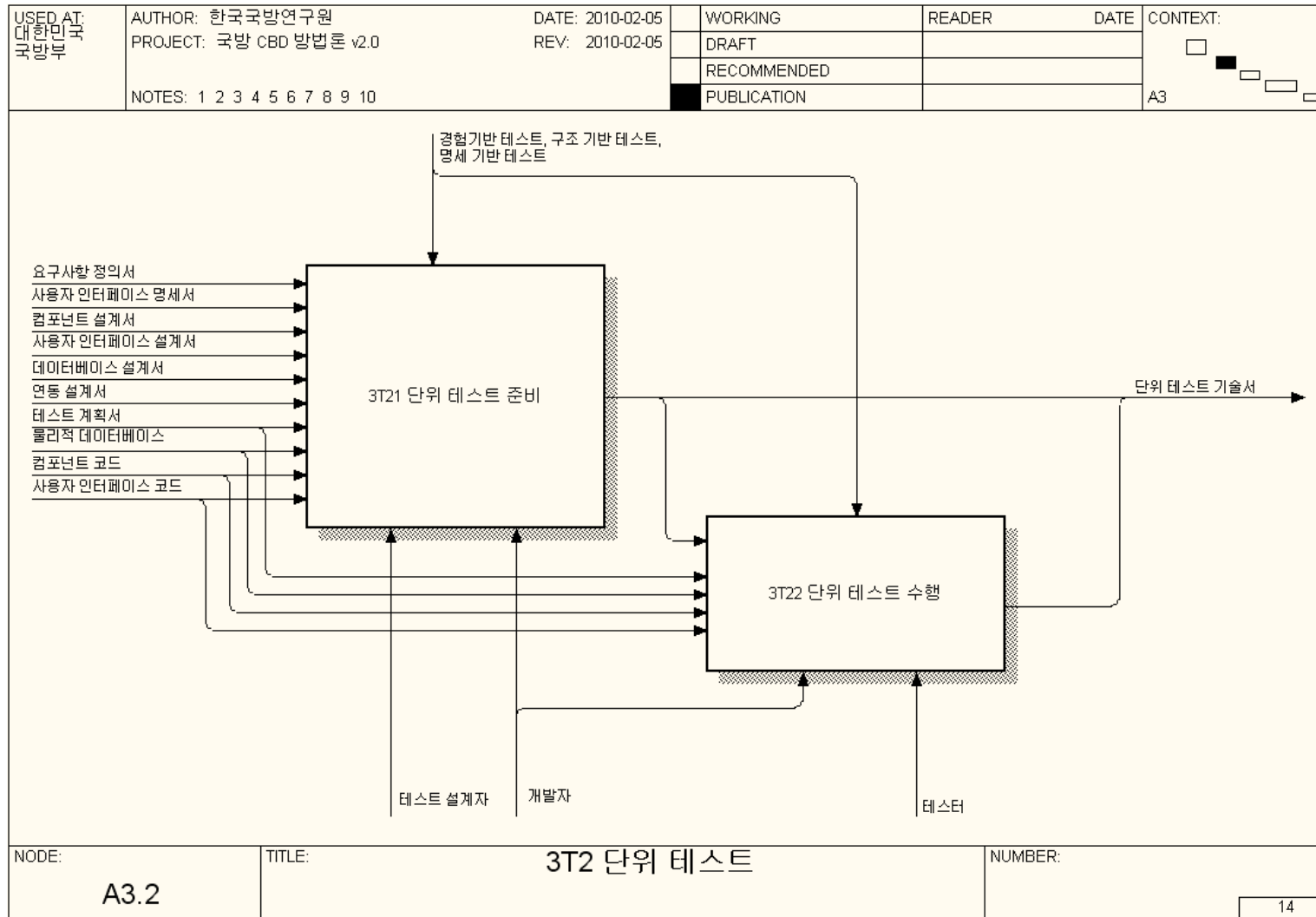
세부 절차



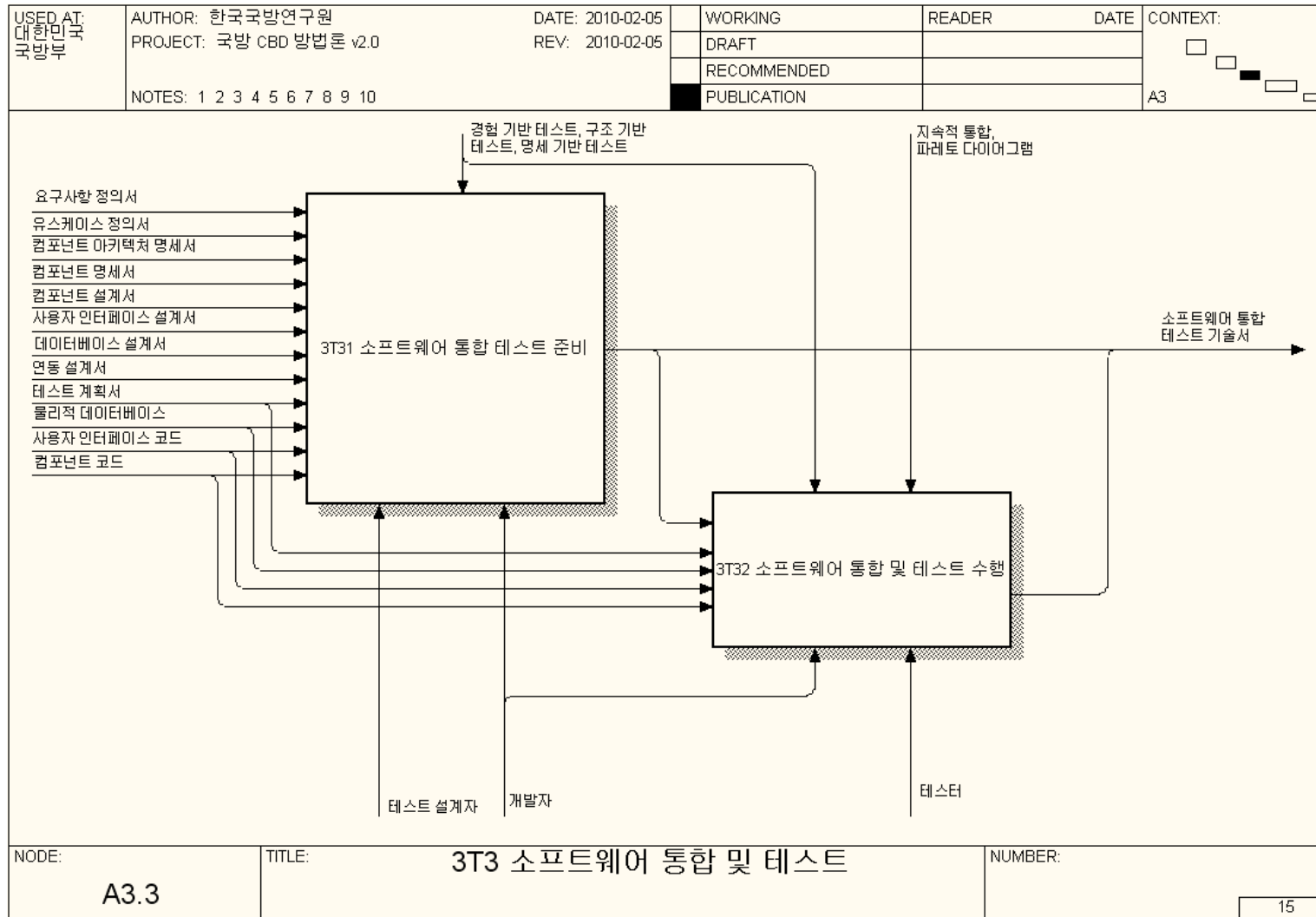
세부 절차



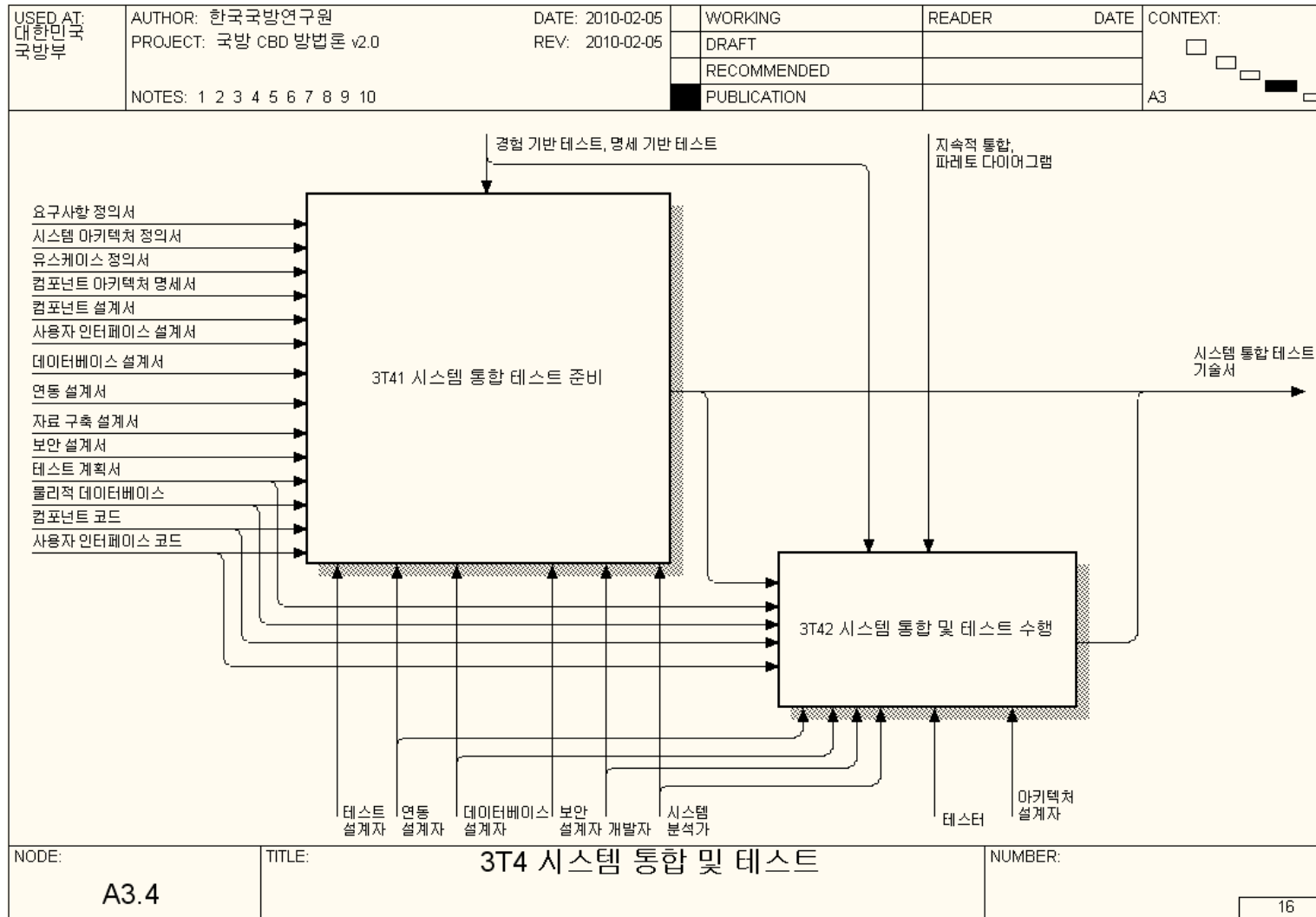
세부 절차



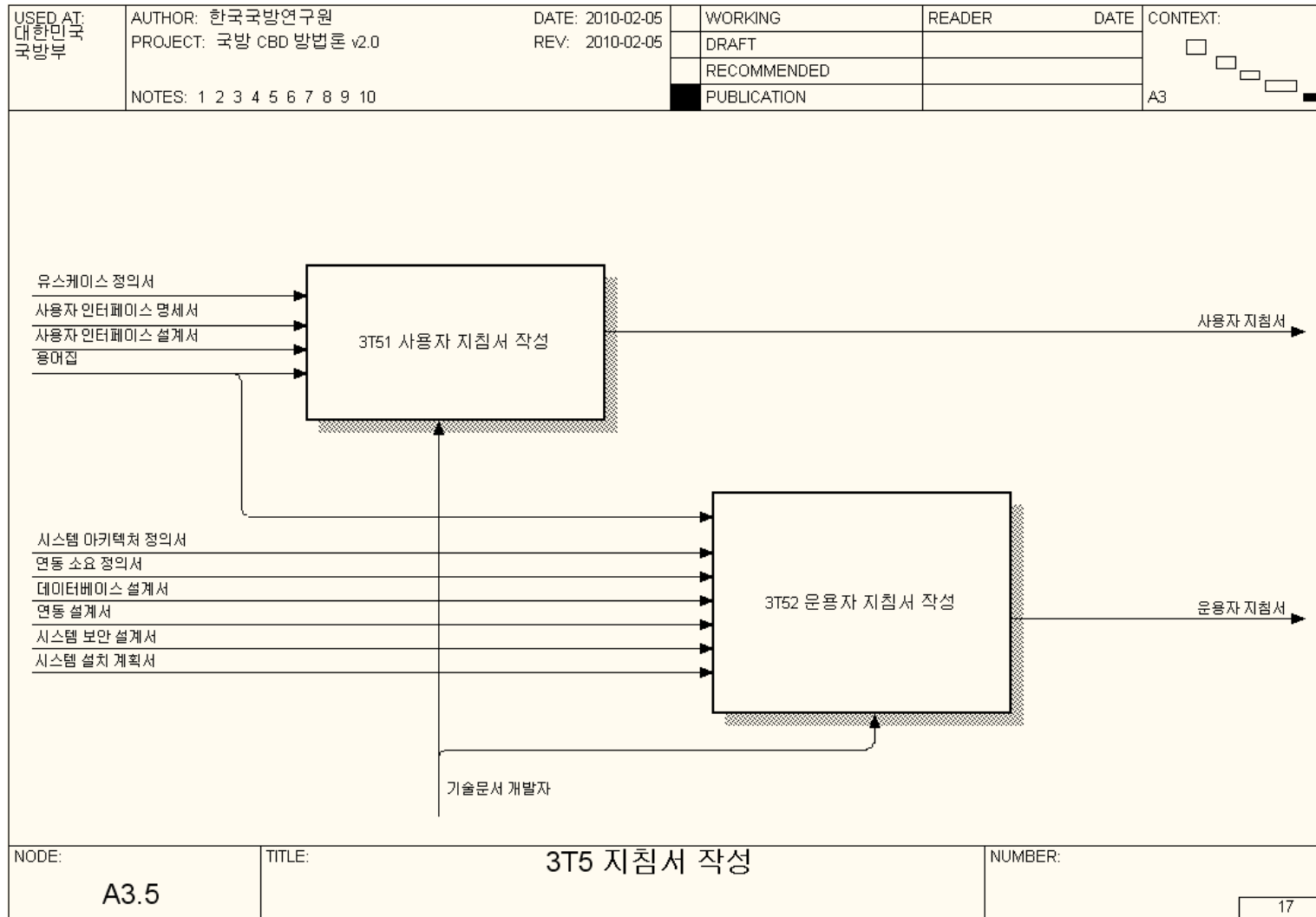
세부 절차



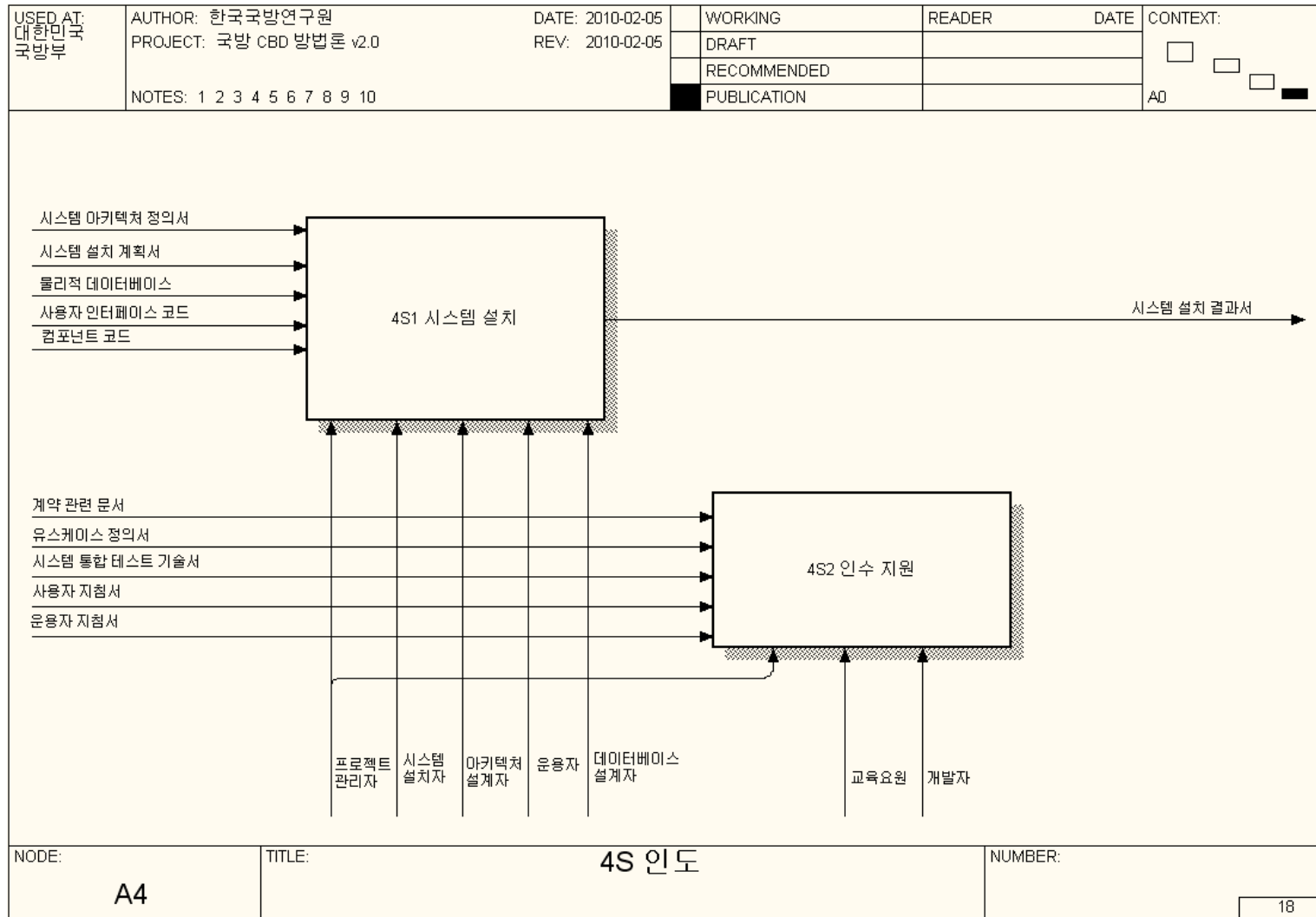
세부 절차



세부 절차



세부 절차



세부 절차

USED AT: 대한민국 국방부	AUTHOR: 한국국방연구원	DATE: 2010-02-05	WORKING	READER	DATE	CONTEXT: <div style="display: inline-block; width: 20px; height: 15px; background-color: black;"></div> <div style="display: inline-block; width: 20px; height: 15px; border: 1px solid black;"></div> A4
	PROJECT: 국방 CBD 방법론 v2.0	REV: 2010-02-05	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						

```

graph LR
    A[시스템 설치 계획서] --> D[4S1 시스템 설치 실시]
    B[시스템 아키텍처 정의서] --> D
    C[물리적 데이터베이스] --> D
    E[사용자 인터페이스 코드] --> D
    F[컴포넌트 코드] --> D
    D --> G[시스템 설치 결과서]
    H[프로젝트 관리자] --> D
    I[시스템 설치자] --> D
    J[아키텍처 설계자] --> D
    K[운영자] --> D
    L[데이터베이스 설계자] --> D
  
```

The diagram illustrates the 4S1 system installation process. It features a central box labeled "4S1 시스템 설치 실시" (4S1 System Installation Implementation). Five input documents/elements point to this central box: "시스템 설치 계획서" (System Installation Plan), "시스템 아키텍처 정의서" (System Architecture Definition), "물리적 데이터베이스" (Physical Database), "사용자 인터페이스 코드" (User Interface Code), and "컴포넌트 코드" (Component Code). Five roles point to the bottom of the central box: "프로젝트 관리자" (Project Manager), "시스템 설치자" (System Installer), "아키텍처 설계자" (Architecture Designer), "운영자" (Operator), and "데이터베이스 설계자" (Database Designer). An output arrow points from the right side of the central box to "시스템 설치 결과서" (System Installation Result Report).

NODE: A4.1	TITLE: 4S1 시스템 설치	NUMBER: <div style="float: right; border: 1px solid black; padding: 2px;">19</div>
----------------------	-----------------------------	---

세부 절차

USED AT: 대한민국 국방부	AUTHOR: 한국국방연구원	DATE: 2010-02-05	WORKING	READER	DATE	CONTEXT: A4
	PROJECT: 국방 CBD 방법론 v2.0	REV: 2010-02-05	DRAFT			
			RECOMMENDED			
			PUBLICATION			
NOTES: 1 2 3 4 5 6 7 8 9 10						


```

graph TD
    A[계약 관련 문서] --> E[4S21 인수지원 실시]
    B[유스케이스 정의서] --> E
    C[시스템 통합 테스트 기술서] --> E
    D[사용자 지침서] --> E
    F[운용자 지침서] --> E
    G[프로젝트 관리자] --> E
    H[교육요원] --> E
    I[개발자] --> E
  
```

NODE: A4.2	TITLE: 4S2 인수 지원	NUMBER: <div>20</div>
----------------------	----------------------------	--------------------------

Tailoring Guide

테일러링 가이드

테일러링 가이드

개요

국방 CBD 개발 방법론은 컴포넌트를 기반으로 소프트웨어를 개발하고 시스템을 구축하기 위한 방법론이며, 그렇지 않은 경우는 적용되지 않는다. 컴포넌트를 기반으로 시스템을 구축하더라도 모든 프로젝트에 동일한 프로세스와 산출물을 적용하는 것은 비효율적일 수 있다. 각각의 프로젝트는 시스템의 규모, 개발기간 및 개발유형 등 다양한 특성을 가지고 있기 때문에 이러한 상황을 고려하지 않은 개발 방법론의 표준 프로세스를 모든 프로젝트에 적용하기 보다는 상황에 맞춘 프로세스로 테일러링하여 적용하는 것이 바람직하다.

이에, 본 가이드는 국방 CBD 방법론을 적절하게 조정하기 위한 테일러링 방법에 대해 제시하고자 한다.

테일러링의 의미

테일러링이란, 사전적 정의로는 양복을 입는 사람의 체형과 요건에 맞게 재봉하는 것으로 '재단하다'라는 뜻을 가지고 있다. 개발 방법론 적용에 있어서 테일러링은 개발하는 시스템의 목적과 특성에 맞게 방법론 프로세스와 프로세스의 세부 내역을 수정, 추가, 삭제하는 모든 것을 의미한다. 즉 각 개발 단계, 활동, 작업에 대한 변경과 산출물 작성 등에 대한 변경 모두를 뜻하는 것이다.

테일러링 방법 및 절차

테일러링은 다음과 같은 방법 및 절차를 통해 적용된다. 세부적인 내용은 다음과 같다.

● 프로젝트 환경의 특성 식별

테일러링에 영향을 미치는 프로젝트 환경의 특성을 식별한다. 이러한 특성의 예로는 시스템의 규모, 개발기간과 개발유형, 시스템의 수명주기 활동 등이 될 수 있다.

● 테일러링 결정과 문서화

식별된 특성을 토대로 테일러링 적용 기준을 결정한다. 가이드에서 제공되는 테일러링 기준을 적용할 수도 있고, 경우에 따라서는 해당 기준을 수정, 보완하거나 새로운 기준을 생성하여 적용할 수도 있다. 다만, 이런 모든 과정 및 내용은 사용자와의 협의를 통해 진행되어야 한다.

테일러링 적용 기준이 결정되면 해당 결정사항과 논리적인 근거에 대해서 별도로 테일러링 정의서를 작성하거나 사업수행계획서 등에 포함해 문서화해야 한다.

● 테일러링 적용 및 감시

프로젝트 진행 기간 동안에 테일러링을 통해 결정된 사항들이 제대로 적용되고 있는지를 분석하고 감시하여야 한다.

테일러링 가이드

테일러링 적용 기준

본 가이드에서는 시스템 개발 유형, 유지보수 수행 및 분리발주 등에 따른 테일러링 적용 기준을 제시한다. 제시되는 테일러링 적용 기준은 절대적인 것이 아니고 하나의 테일러링의 예로서 적용 또는 참조될 수 있으며, 프로젝트의 환경의 특성에 따라 추가적으로 여러 항목을 고려하여 개발 대상 시스템에 적합한 프로세스를 정의하여 적용하는 것이 바람직하다.

1. CBSD 및 CD에 따른 테일러링 가이드

- CBSD(Component Based System Development)는 컴포넌트를 기반으로 시스템을 개발하는 것으로, 이미 개발된 컴포넌트나 새로 개발된 컴포넌트를 활용하여 시스템을 개발하는 것이다.
- CD(Component Development)는 CBSD에서 필요한 컴포넌트를 개발하는 것이다.
- 따라서 CD 수행 사업에 국방 CBD 방법론을 그대로 적용하기는 어려우므로 별도의 방법론을 적용하거나 국방 CBD 방법론의 컴포넌트 관련 산출물 위주로 테일러링하여 적용해야 한다.
- 또한, 공통 컴포넌트에 대해서는 별도의 컴포넌트 저장소를 통해 관리될 계획이므로 그에 따라 요구되는 산출물이나 작업이 있는지 확인하여야 한다.

2. 시스템 규모와 개발기간에 따른 테일러링 가이드

- 시스템 규모와 개발기간에 따라 시스템을 대규모, 소규모로 나누고 각 규모별로 필수 산출물 및 선택 산출물을 정의한다.
- 일반적으로 시스템 규모를 식별하는 절대적인 지침 또는 근거가 존재하지는 않는다. 따라서 국방 CBD 방법론에서는 프로젝트 사업금액, 소프트웨어 개발비 및 개발기간에 따라 규모를 식별하는 기준을 다음과 같이 정의한다.

구분	규모 식별 기준
사업금액	“소프트웨어 산업 진흥법 제24조 2”의 세부 지침인 ‘대기업인 소프트웨어 사업자가 참여할 수 있는 사업 금액의 하한’ 금액 (‘10. 1. 1. 현재 기준 20억 원, 향후 고시 개정에 따라 변동)
소프트웨어 개발비	“전자정부법 시행령”에서 명시된 감리 대상 기준이 되는 소프트웨어 개발 금액 (‘10. 1. 1. 현재 기준 5억 원, 향후 시행령 개정에 따라 변동)
개발기간	시스템 개발기간 1년

테일러링 가이드

- 위의 기준을 토대로 대규모에 해당되는 시스템은 3가지 기준을 모두 상회(사업금액이 20억 원, 소프트웨어 개발비 5억 원, 수행기간이 1년 이상)해야 하며 그 외의 경우는 소규모에 해당한다. 사례에 따른 시스템 규모 식별 결과는 다음과 같다.

기준	사 례							
사업금액 20억 원 이상	○	×	○	×	○	×	○	×
소프트웨어 개발비 5억 원 이상	○	○	×	×	○	○	×	×
개발기간 1년 이상	○	○	○	○	×	×	×	×
규모 식별 결과	대규모	소규모	소규모	소규모	소규모	소규모	소규모	별도고려

- 대규모로 식별된 시스템의 경우, 국방 CBD 방법론에서 제시하고 있는 모든 공정을 준수하여 산출물을 작성하도록 한다. 다만 제시된 공정의 일부가 해당 사업의 과업에 포함되어 있지 않다면 해당 공정을 수행하지 아니하고 산출물 또한 작성하지 않는다.
- 소규모 시스템의 경우는 최소한의 필수 산출물을 기준으로 필요한 공정 또는 작업을 추가로 수행한다. 필수 산출물을 지정한 기준은 다음과 같다.
 - 컴포넌트 기반 개발 방법론으로서 유지해야 하는 최소의 개발 공정
 - 개발 시 개발자들이 지속적으로 참조하며 관리되어야 하는 산출물
 - 사용자 또는 운용자에게 배포되어야 하는 산출물
- 그 외의 시스템의 경우는 추가적인 고려가 필요하다. 일차적으로 규모를 식별하는 3가지 기준에 해당되지 않는 시스템은 소규모보다도 작은 시스템에 해당되기 때문에 프로세스 준수와 많은 산출물 작성을 요구하는 국방 CBD 방법론 적용여부에 대해서 재검토가 필요하다.
- 또한, 소규모보다 작은 시스템 개발에 국방 CBD 방법론을 적용하더라도 작업, 산출물 그리고 수행절차 등에서 많은 부분을 통합하는 것에 대해 고려해야 하며 이는 사업에 특성에 맞게 판단되어야 한다.
- 예를 들어 시스템 개발에 적용되는 표준이 적고, 시스템 아키텍처가 간단하다면 표준 지침서와 시스템 아키텍처 정의서의 통합을 고려할 수 있으며, 분석과 설계단계로 나누어 작성되는 자료구축 또는 연동 관련 산출물 등에 대해 하나로 통합하여 작성하는 것을 고려해 볼 수 있다.
- 시스템 규모 및 개발기간에 따른 방법론 테일러링 세부 가이드는 다음 표와 같다.

테일러링 가이드

● : 필수, ○ : 선택, ■ : 연동/자료구축 시 필수

단계	활동	작업	산출물	대규모	소규모
1R 분석	1R1 요구사항 식별	1R11 도메인 모델링	1R11a 도메인 정의서	●	○
			1R11b 용어집	●	●
		1R12 현행 시스템 분석	1R12a 현행 시스템 분석서	●	○
		1R13 비즈니스 모델링	1R13a 비즈니스 정의서	●	○
	1R2 아키텍처 정의	1R14 요구사항 정의	1R14a 요구사항 정의서 부록 : 요구사항 추적표	●	●
		1R21 시스템 아키텍처 정의	1R21a 시스템 아키텍처 정의서	●	●
	1R3 요구사항 분석	1R22 표준 지침 수립	1R22a 표준 지침서	●	●
		1R31 유스케이스 모델링	1R31a 유스케이스 정의서	●	●
		1R32 연동 소요 분석	1R32a 연동 소요 정의서 부록 : 연동 항목 소요서(IER)	■	■
		1R33 사용자 인터페이스 정의	1R33a 사용자 인터페이스 정의서	●	○
	1R4 자료 구축 준비	1R34 클래스 모델링	1R34a 클래스 정의서	●	●
		1R41 자료 구축 계획	1R41a 자료 구축 계획서	■	■
2D 설계	2D1 개략 설계	1R5 테스트 준비	1R51 테스트 계획	●	○
		2D11 컴포넌트 식별	2D11a 컴포넌트 아키텍처 명세서	●	●
		2D12 인터페이스 상호작용 명세	2D12a 인터페이스 상호작용 명세서	●	○
		2D13 컴포넌트 명세	2D13a 컴포넌트 명세서	●	○
		2D14 사용자 인터페이스 명세	2D14a 사용자 인터페이스 명세서	●	○
	2D2 상세 설계	2D15 데이터 모델링	2D15a 데이터 명세서	●	○
		2D21 컴포넌트 구현 설계	2D21a 컴포넌트 설계서	●	●
			2D21b 트랜잭션 설계서	●	●
		2D22 사용자 인터페이스 구현 설계	2D22a 사용자 인터페이스 설계서	●	●
		2D23 데이터베이스 설계	2D23a 데이터베이스 설계서 부록 : 코드 설계서	●	●
		2D24 연동 설계	2D24a 연동 설계서 부록 : 연동 항목 테이블 매핑 설계서	■	■
		2D25 자료 구축 설계	2D25a 자료 구축 설계서	■	■

테일러링 가이드

단계	활동	작업	산출물	대규모	소규모
		2D26 시스템 보안 설계	2D26a 시스템 보안 설계서	●	●
	2D3 시스템 설치 준비	2D31 시스템 설치 계획	2D31a 시스템 설치 계획서	●	○
3T 구현 및 테스트	3T1 구현	3T11 데이터베이스 구축	3T11a 물리적 데이터베이스	●	●
		3T12 컴포넌트 구현	3T12a 컴포넌트 코드	●	●
		3T13 사용자 인터페이스 구현	3T13a 사용자 인터페이스 코드	●	●
	3T2 단위 테스트	3T21 단위 테스트 준비	3T21a 단위 테스트 기술서	●	○
		3T22 단위 테스트 수행			
	3T3 소프트웨어 통합 및 테스트	3T31 소프트웨어 통합 테스트 준비	3T31a 소프트웨어 통합 테스트 기술서	●	●
		3T32 소프트웨어 통합 및 테스트 수행			
	3T4 시스템 통합 및 테스트	3T41 시스템 통합 테스트 준비	3T41a 시스템 통합 테스트 기술서	●	●
		3T42 시스템 통합 및 테스트 수행			
	3T5 지침서 작성	3T51 사용자 지침서 작성	3T51a 사용자 지침서	●	●
		3T52 운용자 지침서 작성	3T52a 운용자 지침서	●	●
4S 인도	4S1 시스템 설치	4S11 시스템 설치 실시	4S11a 시스템 설치 결과서	●	○
	4S2 인수 지원	4S21 인수 지원 실시	-	-	-
4단계	15활동	37작업	35산출물 4부록	-	-

3. 상용 소프트웨어 패키지 도입에 따른 테일러링 가이드

- 시스템을 신규로 개발하지 않고 현재 상용되고 있는 소프트웨어를 패키지 형태로 도입하는 경우 일반적인 개발 방법론상의 SW 설계 및 구현 단계가 이미 완료되었다고 볼 수 있다. 반면에, 도입된 소프트웨어를 현재 업무에 알맞게 맞추는 단계가 추가되어야 한다.
- 따라서 상용 소프트웨어 패키지를 도입하는 경우, 국방 CBD 방법론의 분석 활동의 일부 작업과 개략설계, 상세설계, 구현 및 단위 테스트 단계의 활동이 생략 가능하다.
- 반대로 해당 소프트웨어 패키지 도입에 따라 필요한 별도의 작업 또는 산출물들은 추가되어야 하며, 추가 작업 또는 산출물은 패키지 유형이나 벤더들에 따라 달라질 수 있으므로 프로젝트 초기 단계에 사용자와의 협의를 통해 지정한다.

테일러링 가이드

- 다만 소프트웨어 패키지를 도입하더라도 시스템 내의 추가 기능을 컴포넌트 형태로 개발하는 경우는 국방 CBD 방법론의 개발 공정을 그대로 적용한다.

● : 필수, ■ : 연동/자료구축 시 필수

단계	활동	작업	산출물	패키지 도입 시
1R 분석	1R1 요구사항 식별	1R11 도메인 모델링	1R11a 도메인 정의서 1R11b 용어집	●
		1R12 현행 시스템 분석	1R12a 현행 시스템 분석서	●
		1R13 비즈니스 모델링	1R13a 비즈니스 정의서	●
		1R14 요구사항 정의	1R14a 요구사항 정의서 부록 : 요구사항 추적표	●
		1R21 시스템 아키텍처 정의	1R21a 시스템 아키텍처 정의서	●
	1R2 아키텍처 정의	1R22 표준 지침 수립	1R22a 표준 지침서	●
		1R31 유스케이스 모델링	1R31a 유스케이스 정의서	●
		1R32 연동 소요 분석	1R32a 연동 소요 정의서 부록 : 연동 항목 소요서(IER)	■
		1R33 사용자 인터페이스 정의	1R33a 사용자 인터페이스 정의서	●
		1R34 클래스 모델링	1R34a 클래스 정의서	-
	1R4 자료 구축 준비	1R41 자료 구축 계획	1R41a 자료 구축 계획서	■
	1R5 테스트 준비	1R51 테스트 계획	1R51a 테스트 계획서	●
2D 설계	2D1 개략 설계	2D11 컴포넌트 식별	2D11a 컴포넌트 아키텍처 명세서	-
		2D12 인터페이스 상호작용 명세	2D12a 인터페이스 상호작용 명세서	-
		2D13 컴포넌트 명세	2D13a 컴포넌트 명세서	-
		2D14 사용자 인터페이스 명세	2D14a 사용자 인터페이스 명세서	-
		2D15 데이터 모델링	2D15a 데이터 명세서	-
	2D2 상세 설계	2D21 컴포넌트 구현 설계	2D21a 컴포넌트 설계서 2D21b 트랜잭션 설계서	-
		2D22 사용자 인터페이스 구현 설계	2D22a 사용자 인터페이스 설계서	-
		2D23 데이터베이스 설계	2D23a 데이터베이스 설계서 부록 : 코드 설계서	-
		2D24 연동 설계	2D24a 연동 설계서 부록 : 연동 항목 테이블 매핑 설계서	■

테일러링 가이드

단계	활동	작업	산출물	패키지 도입 시
		2D25 자료 구축 설계	2D25a 자료 구축 설계서	■
		2D26 시스템 보안 설계	2D26a 시스템 보안 설계서	●
	2D3 시스템 설치 준비	2D31 시스템 설치 계획	2D31a 시스템 설치 계획서	●
3T 구현 및 테스트	3T1 구현	3T11 데이터베이스 구축	3T11a 물리적 데이터베이스	-
		3T12 컴포넌트 구현	3T12a 컴포넌트 코드	-
		3T13 사용자 인터페이스 구현	3T13a 사용자 인터페이스 코드	-
	3T2 단위 테스트	3T21 단위 테스트 준비	3T21a 단위 테스트 기술서	-
		3T22 단위 테스트 수행		
	3T3 소프트웨어 통합 및 테스트	3T31 소프트웨어 통합 테스트 준비	3T31a 소프트웨어 통합 테스트 기술서	●
		3T32 소프트웨어 통합 및 테스트 수행		
	3T4 시스템 통합 및 테스트	3T41 시스템 통합 테스트 준비	3T41a 시스템 통합 테스트 기술서	●
		3T42 시스템 통합 및 테스트 수행		
	3T5 지침서 작성	3T51 사용자 지침서 작성	3T51a 사용자 지침서	●
		3T52 운용자 지침서 작성	3T52a 운용자 지침서	●
4S 인도	4S1 시스템 설치	4S11 시스템 설치 실시	4S11a 시스템 설치 결과서	●
	4S2 인수 지원	4S21 시스템 설치 실시	-	-
4단계	15활동	37작업	35산출물 4부록	

테일러링 가이드

4. 유지보수 수행 시에 테일러링 가이드

- 유지보수는 아래 표와 같이 신규개발, 재개발과는 별도로 체계운용과정에서 발생하는 요구사항이나 문제점, 변경에 대한 조치를 수행하는 것으로 개발방법론 전체 프로세스의 일부를 미니 프로세스로 사용하는 것이 일반적이다.

구분	정의
신규개발	새로운 정보체계를 개발하는 것
재개발 (성능 개량)	개발된 소프트웨어의 일부를 다시 개발하는 것으로 업무량 또는 산정된 비용이 유지보수의 범위를 초과하는 경우
유지보수	체계운용과정에서 발생하는 요구사항이나 문제점, 변경에 대한 조치를 수행하는 것

- 이에 따라 국방 CBD 방법론에서는 컴포넌트 기반 개발 방법론으로서 유지해야 하는 최소의 개발 공정인 소규모 체계의 테일러링 가이드를 기준으로 시스템의 최종 모습이 담겨있고, 사용자와 개발자가 지속적으로 참조하는 산출물을 선별하여 유지보수 시 반드시 갱신하고 관리되어야 할 대상으로 정의한다.
- 재개발의 경우 기존 기능 수정과 신규 기능 개발이 같이 이루어지므로 단순히 유지보수 테일러링 기준을 적용하기는 어렵다.
- 따라서 재개발의 경우 추가, 개선 및 수정사항이 원활히 반영되고 산출물의 관리가 용이할 수 있도록 신규개발에 준해 방법론을 테일러링하여 적용하는 것이 타당할 것이다. 다만, 기존 개발 산출물을 최대한 활용하여 작성하도록 한다.

● : 필수, ■ : 연동/자료구축 시 필수

단계	활동	작업	산출물	유지보수 시
1R 분석	1R1 요구사항 식별	1R11 도메인 모델링	1R11a 도메인 정의서 1R11b 용어집	- ●
		1R12 현행 시스템 분석	1R12a 현행 시스템 분석서	-
		1R13 비즈니스 모델링	1R13a 비즈니스 정의서	-
		1R14 요구사항 정의	1R14a 요구사항 정의서 부록 : 요구사항 추적표	●
	1R2 아키텍처 정의	1R21 시스템 아키텍처 정의	1R21a 시스템 아키텍처 정의서	●
		1R22 표준 지침 수립	1R22a 표준 지침서	●
	1R3 요구사항 분석	1R31 유스케이스 모델링	1R31a 유스케이스 정의서	●
		1R32 연동 소요 분석	1R32a 연동 소요 정의서 부록 : 연동 항목 소요서(IER)	■
		1R33 사용자 인터페이스 정의	1R33a 사용자 인터페이스 정의서	-
		1R34 클래스 모델링	1R34a 클래스 정의서	-

테일러링 가이드

단계	활동	작업	산출물	유지보수 시
	1R4 자료 구축 준비	1R41 자료 구축 계획	1R41a 자료 구축 계획서	-
	1R5 테스트 준비	1R51 테스트 계획	1R51a 테스트 계획서	-
2D 설계	2D1 개략 설계	2D11 컴포넌트 식별	2D11a 컴포넌트 아키텍처 명세서	●
		2D12 인터페이스 상호작용 명세	2D12a 인터페이스 상호작용 명세서	-
		2D13 컴포넌트 명세	2D13a 컴포넌트 명세서	-
		2D14 사용자 인터페이스 명세	2D14a 사용자 인터페이스 명세서	-
		2D15 데이터 모델링	2D15a 데이터 명세서	-
	2D2 상세 설계	2D21 컴포넌트 구현 설계	2D21a 컴포넌트 설계서 2D21b 트랜잭션 설계서	● ●
		2D22 사용자 인터페이스 구현 설계	2D22a 사용자 인터페이스 설계서	●
		2D23 데이터베이스 설계	2D23a 데이터베이스 설계서 부록 : 코드 설계서	●
		2D24 연동 설계	2D24a 연동 설계서 부록 : 연동 항목 테이블 매핑 설계서	■
		2D25 자료 구축 설계	2D25a 자료 구축 설계서	-
		2D26 시스템 보안 설계	2D26a 시스템 보안 설계서	-
	2D3 시스템 설치 준비	2D31 시스템 설치 계획	2D31a 시스템 설치 계획서	-
3T 구현 및 테스트	3T1 구현	3T11 데이터베이스 구축	3T11a 물리적 데이터베이스	●
		3T12 컴포넌트 구현	3T12a 컴포넌트 코드	●
		3T13 사용자 인터페이스 구현	3T13a 사용자 인터페이스 코드	●
	3T2 단위 테스트	3T21 단위 테스트 준비	3T21a 단위 테스트 기술서	-
		3T22 단위 테스트 수행		
	3T3 소프트웨어 통합 및 테스트	3T31 소프트웨어 통합 테스트 준비	3T31a 소프트웨어 통합 테스트 기술서	-
		3T32 소프트웨어 통합 및 테스트 수행		
	3T4 시스템 통합 및 테스트	3T41 시스템 통합 테스트 준비	3T41a 시스템 통합 테스트 기술서	-
		3T42 시스템 통합 및 테스트 수행		
	3T5 지침서 작성	3T51 사용자 지침서 작성	3T51a 사용자 지침서	●
		3T52 운용자 지침서 작성	3T52a 운용자 지침서	●
4S 인도	4S1 시스템 설치	4S11 시스템 설치 실시	4S11a 시스템 설치 결과서	-
	4S2 인수 지원	4S21 인수 지원 실시	-	-
4단계	15활동	37작업	35산출물 4부록	17산출물 3부록

테일러링 가이드

5. 소프트웨어 수명주기 모델에 따른 테일러링 가이드

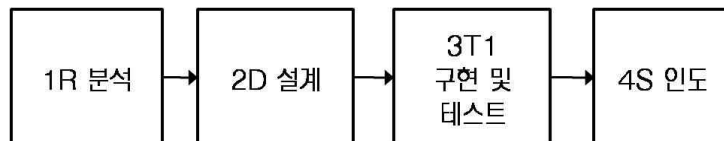
- ‘소프트웨어사업 관리감독에 관한 일반기준(정보통신부고시 제2006-39호)’의 테일러링 지침에서 언급된 것과 같이 기본적인 소프트웨어 생명주기 모델로는 다음과 같이 세 가지의 종류를 고려할 수 있다.

소프트웨어 생명주기 모델	모든 요구사항을 초기에 정의하는가?	복수의 개발 주기를 사용하는가?	중간단계에 소프트웨어를 배치하는가?
일괄 개발 (Waterfall)	예	아니오	아니오
점증적 개발 (Incremental)	예	예	예
진화적 개발 (Evolutionary)	아니오	예	예

- 소프트웨어 생명주기 모델에 따라 방법론은 다르게 적용될 수 있으며 세부적인 내용은 아래의 내용을 참조한다.

● 일괄 개발 전략

일괄 개발 전략은 본질적으로 처음부터 끝까지 단 한번에, 각각의 단계는 단 한번만 수행하는 전략이다. 즉, 사용자의 요구를 결정하고, 요구사항을 정의하며, 시스템을 설계하고, 시스템을 구현하며, 시험하고, 납품하는 것이다. 이러한 접근 방법에서 각 소프트웨어 항목이 개발됨에 따라 개발 프로세스에 활동 및 작업들은 일반적으로 순차적으로 채택된다.



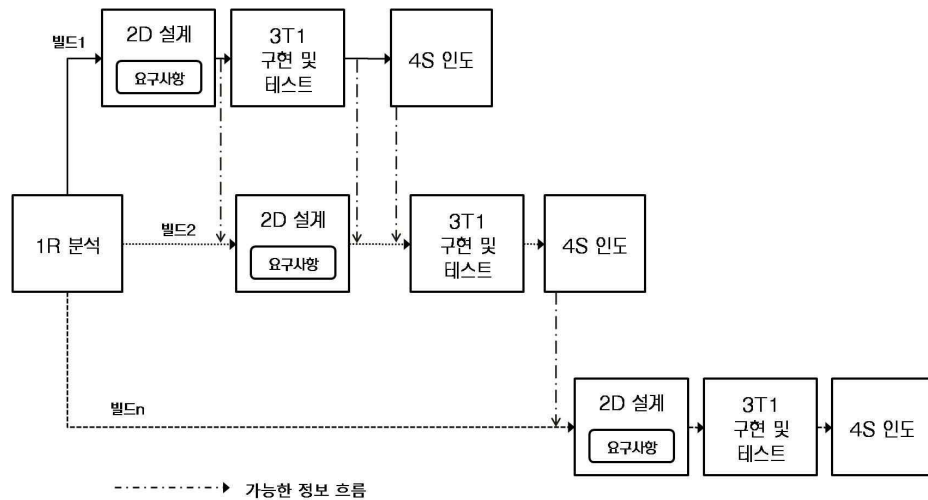
● 점증적 개발 전략

점증적 개발 전략은 일련의 주어진 요구사항으로부터 출발하여 순차적인 빌드과정을 통해 개발한다. 첫 번째 빌드에서 요구사항의 일부를 통합하고 다음번에 추가 요구사항을 통합하는 과정을 시스템이 완성될 때까지 계속적으로 해 나가는 것이다. 각 빌드에서 필요로 하는 프로세스, 활동, 작업들이 수행된다. 예를 들면, 요구분석 및 시스템 아키텍처 정의는 한 번 수행해도 되지만 소프트웨어 설계, 구현 및 테스트는 각 빌드마다 수행된다.

이러한 접근 방법에서 각 빌드가 이루어짐에 따라 개발 프로세스에서의 활

테일러링 가이드

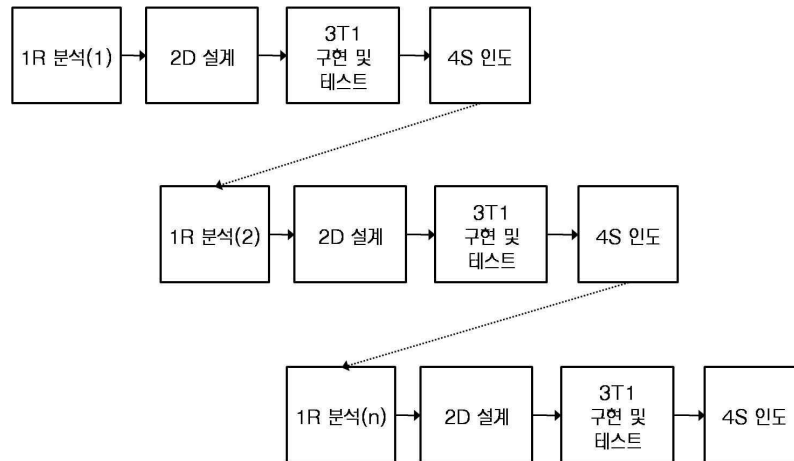
동 및 작업이 순차적 또는 중복되어 부분적으로 동시에 채택된다. 일련의 빌드과정에서 부분적으로 동시에 개발될 때 개발 프로세스의 활동 및 작업들을 빌드과정에서 동시에 채택해도 된다. 개발 프로세스의 활동과 작업들은 모든 빌드과정에서 똑같은 순서를 가지고 반복적으로 이루어지는 것이 일반적이다.



● 진화적 개발 전략

진화적 개발 전략은 점진적 개발과 마찬가지로 빌드를 통하여 시스템을 개발하지만, 모든 요구사항을 처음부터 정의할 수 없다는 점에서 점증적 개발과 다르다. 진화적 개발 전략에서, 사용자 및 시스템 요구사항이 처음에는 부분적으로만 정의되며, 후속적인 빌드가 진행됨에 따라 자세하게 정의된다. 이러한 접근 방법에서 각 빌드과정이 진행됨에 따라 개발 프로세스에서 활동과 작업들은 순차적으로 또는 부분적으로 중복되어 채택된다. 개발 프로세스의 활동과 작업들은 모든 빌드 과정에서 똑같은 순서를 가지고 반복적으로 이루어지는 것이 일반적이다.

테일러링 가이드



6. 국방 도메인에 따른 테일러링 가이드

- 일반적으로 업무 도메인에 따라 개발 방법론을 다르게 적용하지는 않는다. 다만, 업무 도메인이 다르기 때문에 프로젝트의 특성에서 차이가 존재하고 그에 따라 방법론의 테일러링이 달라지는 것이다.
- 국방 정보체계는 크게 자원관리 정보체계와 전장관리 정보체계로 나눌 수 있다(기반 체계 제외). 만일 컴포넌트 기반으로 시스템을 개발하도록 전략이 수립되면 두 가지 모두 국방 CBD 방법론을 적용한다.
- 다만, 전장관리의 경우 그 특성상 소프트웨어가 대규모의 하드웨어, 정보 자산, 무기체계 등과 통합되는 것이 중시되며 프로젝트 과업의 상당 비율을 차지하고 있다.
- 따라서 이와 같은 경우 컴포넌트 기반의 소프트웨어 개발 부분만 국방 CBD 방법론을 적용하고 그 외 부분은 해당 특성에 맞도록 방위사업청의 개발 프로세스를 적용하는 것으로 테일러링 기준을 수립할 수 있다.
- 내장형 소프트웨어도 컴포넌트 기반의 소프트웨어 개발일 경우 국방 CBD 방법론을 사업에 맞게 테일러링하여 적용하고 그렇지 않은 경우 별도의 적합한 방법론을 선정하여 적용한다.
- 이와 같이 국방 도메인에 따른 시스템 특성에 맞게 국방 CBD 방법론의 공정 및 산출물의 추가, 보완이 필요한 경우, 적절한 테일러링 기준을 세우고 이를 사용자와 협의를 통해 결정해야 한다.

7. SW 분리/분할발주에 따른 테일러링 가이드

● SW 분리발주

- SW 분리발주는 HW, SW, 시스템 통합 등을 일괄하여 계약하지 않고 일정 기준 이상 규모(‘소프트웨어 산업 진흥법’ 등에 따른 대상 지정 고시)의 소프트웨어를 분리하여 발주·계약하는 형태를 의미한다. 정부는 공공 SW사업의 품질 향상 및 SW기업의 전문화를 통한 SW산업의 경쟁력 재고 차원에서 SW분리발주를 권장하고 있다.
- SW 분리발주는 SW사업 수행 시 분리발주 대상 SW를 정하기 위해 구체적으로 시스템을 분석하고 공개경쟁에 의해 기술성 평가를 수행하여 우수 SW의 선정이 가능하게 한다.
- SW 분리발주에 따라 국방 CBD 방법론의 프로세스 및 수행내용에 변화가 생기지는 않는다. 다만, SW 분리발주에 따라 시스템에 도입되는 SW가 언제 결정되는가에 따라 1R21 시스템 아키텍처 정의 작업에서 고려해야 할 사항에 변화가 생긴다.
- 도입되는 SW가 시스템 아키텍처 정의 작업 이전에 결정되면 해당 SW에 맞게 아키텍처를 정의할 수 있지만, 이후에 결정된다면 시스템에 적합한 SW가 도입될 수 있도록 세부적인 기능 및 요구사항을 시스템 아키텍처 정의 작업에서 도출해야 한다.

● SW 분할발주

- SW 분할발주는 시스템 분석, 설계, 구현 등 각 단계를 분리하여 발주·계약하는 형태를 의미하며 원칙적으로 설계를 완벽하게 하고 이를 바탕으로 구현 프로젝트를 발주하게 되면 수행 업무량과 비용의 불확실성이 감소되어 품질과 프로젝트의 성공 확률이 제고된다는 측면이 고려되어 있다.
- 실제 SW 분할발주를 고려할 경우 개발 대상 시스템의 난이도, SW 개발 인력의 개발 성숙도 수준, 의사소통 및 품질 검증 비용 등을 감안해야 한다.
- 국방 CBD 방법론 측면에서는 SW분리발주와 마찬가지로 프로세스 및 수행내용에 변화가 생기지는 않는다.
- 특히 기존 국방 CBD 방법론 v1.1에서는 사용자 인터페이스에 대한 작업이 분석 및 설계에 나누어 있어 사용자 인터페이스를 통한 요구사항 확정이 어려운 면이 있었으나 v2.0에서는 1R33 사용자 인터페이스 정의 작업에서 모든 화면에 대한 검토를 진행함으로써 분석 단계와 설계 단계의 분할발주가 용이한 측면이 있다.
- 다만, 방법론에서 제시되는 개발 단계를 SW 분할발주의 기준으로 고려해

테일러링 가이드

야 하며 단계를 나누어서 발주·개발함에 따라 선행 사업에서 작성한 산출물에 대한 확인 작업 및 현행화 문제가 발생할 수 있다.

- 후행 사업에서 선행 사업의 작업을 재수행하고 산출물을 재작성하는 것은 SW 분할발주 취지에 적합하지 않는다. 다만, 변경사항에 대해서는 수정 보완이 필요하며 그 대상 산출물을 최소화 할 필요가 있다.
- SW 분리발주에 비해 SW 분할발주는 아직까지 도입에 따른 세부적인 작업 지침이나 가이드가 제시되어 있지 않다. 향후, 정부 시책에 따라 세부 작업 지침이나 가이드가 고시되면 이를 참조한다.

8. 작업별 테일러링 가이드

- 프로젝트의 특성에 따라 작업 또는 산출물의 생략, 산출물 작성 시 유의사항 등에 대해 고려할 필요가 있으며 이에 작업별 테일러링 가이드를 별도로 제시한다.
- 작업별 테일러링 가이드는 다른 테일러링 기준을 적용한 결과에 추가적으로 적용할 수 있으며, 세부 내용은 각 작업설명서를 참조한다.

● 1R 분석 단계

활동	작업	작업별 테일러링 가이드
1R1 요구사항 식별	1R11 도메인 모델링	<ul style="list-style-type: none"> • 개발 대상 시스템의 도메인과 상호작용하는 시스템 외부 객체가 이미 명확히 식별되어 있고 국방정보기술아키텍처 등에서 작성한 도메인 모델 정의의 관련 산출물이 존재하며 세부 내용에 차이가 없는 경우 해당 산출물로 작업 수행을 대체할 수 있다.
	1R12 현행 시스템 분석	<ul style="list-style-type: none"> • 현행 시스템 분석 작업은 분석 대상의 선정이 중요하므로 개발자가 현행 시스템 대상을 식별한 후 사용자와의 협의를 통해 수행 여부를 결정한다. • 사전에 획득한 현행 관련 문서 또는 현행 시스템의 개발 산출물이 존재하고, 그것을 통해 충분한 분석이 가능한 경우 사용자와의 협의를 통해 관련 문서로 대체할 수 있다. • 현행 시스템 분석서의 서식은 국방아키텍처프레임워크(MND-AF) v1.2에서 제시된 산출물의 항목들과 많은 부분이 일치한다. 이는 국방정보기술아키텍처의 AS-IS 아키텍처를 활용하기 위한 것으로 현행 시스템이 국방정보기술아키텍처로 구축된 경우 체계관점 산출물들을 참조하여 산출물을 작성한다.

테일러링 가이드

활동	작업	작업별 테일러링 가이드
		<ul style="list-style-type: none"> 현행 시스템과 관련한 기존 산출물이 미흡할 경우, 제시된 현행 시스템 분석서 서식을 모두 적용하지 않아도 무방하다. 다만, 본 작업은 현행 시스템 분석을 통해 운용 과정을 이해하고 현행 시스템의 문제점을 파악하며 그에 대한 해결방안을 도출하기 위한 것으로 목적에 맞게 산출물을 재구성하여 작성한다. 현행 시스템의 테이블과 코드 등에 관련된 자료가 별도로 존재하는 경우 현행 시스템 분석서 서식의 데이터베이스 구성 항목을 대체할 수 있다.
	1R13 비즈니스 모델링	<ul style="list-style-type: none"> 비즈니스 모델링 작업은 개발 대상 시스템이 지원해야 할 업무 범위를 명확히 하고 업무 수행에 따른 주요 행위자, 업무 처리 흐름, 입출력 정보를 정의하는데 목적이 있으므로 수집한 자료가 위의 사항을 만족하면 기존 자료를 참고하는 것으로 비즈니스 모델링 작업을 대체할 수 있다.
	1R14 요구사항 정의	<ul style="list-style-type: none"> 부록인 요구사항 추적표는 정형화된 서식이 존재하는 것은 아니므로 프로젝트 및 시스템에 특성에 따라 적절한 서식을 정의하여 작성한다. 요구사항 추적표는 요구사항을 기준으로 시스템 개발 과정에서의 추적을 목적으로 하지만 필요시 인도의 용이성이나 사업관리의 용이성을 고려하여 제안 요청서, 제안서, 기술 협상 단계부터 추적할 수도 있다.
1R1 아키텍처 정의	1R21 시스템 아키텍처 정의	<ul style="list-style-type: none"> 패키지 소프트웨어나 상용 프레임워크의 도입, 기존 시스템의 커스터마이징 등에 따라 아키텍처 정의 문서가 별도로 존재할 경우에는 해당 문서를 필요시 별도로 첨부하고, 본 방법론의 시스템 아키텍처 정의서는 공통성을 위해 서식을 최대한 준수하여 작성한다.
	1R22 표준 지침 수립	<ul style="list-style-type: none"> 표준 지침서 서식에서 제시하는 항목 이외에 프로젝트의 특성이나 개발 환경 등 기타 추가적인 표준이 필요하면 추가로 작성하도록 한다. 별도의 표준문서가 존재하는 경우, 해당 표준을 준수할 것을 정의하고 참조하도록 기술한다.
1R3 요구사항 분석	1R31 유스케이스 모델링	해당 사항 없음
	1R32 연동 소요 분석	<ul style="list-style-type: none"> 연동 소요 분석 시 패키지 소프트웨어나 상용 프레

테일러링 가이드

활동	작업	작업별 테일러링 가이드
		<p>임워크의 도입, 기존 시스템의 커스터마이징, 타 시스템의 연동 모듈 제공 등과 같이 연동하기 위한 기술 요건이 이미 정의되었다면 이를 바탕으로 기술하며, 필요시 산출물 서식을 추가하여 작성할 수 있다.</p> <ul style="list-style-type: none"> 해당 기술 적용에 필요한 별도의 분석 단계 산출물 (예 : EAI 표준화 정의서 등)이 존재한다면 부록으로 첨부한다.
	1R33 사용자 인터페이스 정의	<ul style="list-style-type: none"> 유스케이스 설명에 따라 시스템의 구현 및 시스템의 재사용을 고려하여 사용자 화면간의 제어 흐름을 액티비티 다이어그램으로 나타낸다. 주요한 화면의 흐름을 위주로 작성하며 자명한 흐름은 생략 가능하다.(기간선택, 부대선택 등과 같이 반복적이고, 공통적으로 사용되는 화면에 대한 흐름은 생략 가능하다.)
	1R34 클래스 모델링	해당 사항 없음
1R4 자료구축 준비	1R41 자료 구축 계획	<ul style="list-style-type: none"> 데이터 통합 및 전환 작업은 사업 내에서 하나의 프로젝트로 구분될 수 있다. 따라서 이에 맞는 별도의 방법론을 적용하여 절차와 산출물이 상이하다면 자료 구축 계획서의 절차 항목에 이를 기술하고 해당 산출물을 별첨 또는 별권으로 제출한다.
1R5 테스트 준비	1R51 테스트 계획	<ul style="list-style-type: none"> 소규모 사업의 경우 각 테스트 준비 작업과 테스트를 병행하여 실시가 가능하므로 생략이 가능하다.

테일러링 가이드

● 2D 설계 단계

구현 플랫폼을 전제로 시스템의 소요가 제기되었거나 구현 플랫폼에 종속적인 기능이 존재하는 등 설계 이전에 구현 플랫폼이 결정되었을 경우 방법론 테일러링을 통해 개략 설계와 상세 설계 활동을 병행 또는 통합해 수행한다.

활동	작업	작업별 테일러링 가이드
2D1 개략설계	2D11 컴포넌트 식별	<ul style="list-style-type: none"> 재사용할 컴포넌트 선정 시 국방 상호운용성관리 지시에 따라 국방 공통컴포넌트의 활용을 우선적으로 고려해야하며 기능뿐만 아니라, 컴포넌트 아키텍처 상에 문제가 없는지, 기술적 제약 사항은 없는지 확인해야 한다.
	2D12 인터페이스 상호작용 명세	해당 사항 없음
	2D13 컴포넌트 명세	해당 사항 없음
	2D14 사용자 인터페이스 명세	해당 사항 없음
	2D15 데이터 모델링	<ul style="list-style-type: none"> 데이터 모델링 작업은 소규모 사업의 경우 데이터베이스 설계 작업과 병행하고 산출물로 데이터베이스 설계서만 작성할 수 있다. 객체지향 데이터베이스시스템을 사용한다면 본 작업의 절차에 준해 데이터 모델을 객체지향 관점으로 작성한다.
2D2 상세설계	2D21 컴포넌트 구현 설계	<ul style="list-style-type: none"> 프로젝트 요구사항 중 특정 플랫폼의 적용이 명시되어 있거나 프레임워크를 사용해 구현 플랫폼으로의 종속을 제거할 경우 컴포넌트 명세(2D13)와 컴포넌트 구현 설계(2D21)의 작업을 조정해 중복 작업을 최소화하도록 한다.
	2D22 사용자 인터페이스 구현 설계	해당 사항 없음
	2D23 데이터베이스 설계	<ul style="list-style-type: none"> DBMS는 제조사와 제품, 버전에 따라 정의 및 기술되어야 하는 내용이 다양하고 일부 기능을 지원하지 않는 제품이 존재하므로 각 사업의 특성에 맞게 산출물에 해당 내용을 추가로 기술해야 한다. (예 : synonym 등) 데이터베이스가 상용 소프트웨어 또는 자체적 로직에

테일러링 가이드

활동	작업	작업별 테일러링 가이드
		의해 암호화될 경우 암호화 방식, 암호화 대상 등에 대한 내용을 별도의 산출물로 작성하여 데이터베이스 설계서의 부록으로 첨부한다.
	2D24 연동 설계	<ul style="list-style-type: none"> 패키지 소프트웨어나 상용 프레임워크의 도입, 기존 시스템의 커스터마이징, 타 시스템의 연동 모듈 제공 등과 같이 연동 기술 적용에 필요한 별도의 설계 단계 산출물(예 : EAI 네이밍 정의서, EAI 시스템 설계서 등)이 존재할 경우 이를 부록으로 첨부한다.
	2D25 자료 구축 설계	해당 사항 없음
	2D26 시스템 보안 설계	<ul style="list-style-type: none"> 시스템 보안정책 구조 식별 시 국방정보기술아키텍처에서 정의된 보안정책 구조의 적용을 우선적으로 검토한다. 다만, 개발 대상 시스템의 특성에 따라 별도의 보안정책 구조를 수립할 수 있다.
2D3 시스템 설치 준비	2D31 시스템 설치 계획	<ul style="list-style-type: none"> 시스템 설치 계획 작업은 시스템을 운용 환경에 이상 없이 설치하는데 목적이 있으므로 소프트웨어를 하드웨어에 설치된 상태에서 납품하는 등 해당 사업이 시스템 설치에 대한 과업이 포함되어 있지 않거나 운용 환경에 특별한 설치 작업 요소가 없다면 생략될 수 있다.

테일러링 가이드

● 3T 구현 및 테스트 단계

활동	작업	작업별 테일러링 가이드
3T1 구현	3T11 데이터베이스 구축	해당 사항 없음
	3T12 컴포넌트 구현	해당 사항 없음
	3T13 사용자 인터페이스 구현	해당 사항 없음
3T2 단위 테스트	3T21 단위 테스트 준비	<ul style="list-style-type: none"> 단위 테스트 준비 및 수행 작업은 구현된 단위 소프트웨어의 테스트를 설계하고 수행하는 작업으로써 반드시 실시해야 하나 일반적으로 컴포넌트 구현, 사용자 인터페이스 구현, 데이터베이스 구축 활동과 병행해서 실시할 수 있으므로 소규모 사업의 경우 별도로 분리해서 실시하지 않을 수 있다.
	3T22 단위 테스트 수행	
3T3 소프트 웨어 통합 및 테스트	3T31 소프트웨어 통합 테스트 준비	<ul style="list-style-type: none"> 상용 소프트웨어(또는 컴포넌트)의 요구되는 기능 및 능력이 분명하다면 개발된 컴포넌트와 상용 소프트웨어를 통합하기 전, 상용 소프트웨어가 적정한 성능을 구현하는지, 기능이 요구사항에 부합하는지에 대한 테스트를 수행할 수 있다. 만일 상용 소프트웨어를 분리발주 과정의 벤치마크 테스트를 거쳐 도입하였다면 이 과정은 별도로 수행하지 않는다.
	3T32 소프트웨어 통합 및 테스트 수행	해당 사항 없음
3T4 시스템 통합 및 테스트	3T41 시스템 통합 테스트 준비	해당 사항 없음
	3T42 시스템 통합 및 테스트 수행	해당 사항 없음
3T5 지침서 작성	3T51 사용자 지침서 작성	<ul style="list-style-type: none"> 사용자의 수준 및 요구사항에 따라 서식은 보완해서 사용할 수 있다. 사용자 지침서와 교육 자료, 온라인 도움말 등이 각각 작성되어 자료 중복으로 인한 혼란, 갱신 후 각 내용 불일치 등의 문제가 발생되지 않도록 가급적 통합된 용도의 지침서가 작성되도록 해야 한다. 일반 사용자뿐만 아니라 관리자의 사용 지침 및 방법도 기술해야 하며 필요시 지침서를 분권할 수 있다.

테일러링 가이드

활동	작업	작업별 테일러링 가이드
	3T52 운용자 지침서 작성	<ul style="list-style-type: none"> • 운용자 지침서는 시스템을 운용하는 곳의 지침, 개발 시스템의 유형 등의 요소에 따라 서식이 달라질 수 있다.

● 4S 인도 단계

활동	작업	작업별 테일러링 가이드
4S1 시스템 설치	4S11 시스템 설치 실시	<ul style="list-style-type: none"> • 시스템 설치 실시 작업은 소프트웨어만을 납품하는 등 해당 사업이 시스템 설치에 대한 과업이 포함되어 있지 않다면 생략될 수 있다. • 소프트웨어 개발 공정과 하드웨어 등의 기반 요소 납품 및 설치 시점이 일치하지 않을 수 있으므로 모든 설치 대상이 후반부에 한꺼번에 설치되어야 할 필요가 없고 사업에 특성에 맞게 계획을 수립한 후 계획에 맞게 설치될 수 있다.
4S2 인수지원	4S21 인수 지원 실시	해당 사항 없음

Explanation for Procedures & Template

절차 및 서식 설명

분석 단계(1R)	1
도메인 모델링 작업(1R11)	4
현행 시스템 분석 작업(1R12)	11
비즈니스 모델링 작업(1R13)	34
요구사항 정의 작업(1R14)	44
시스템 아키텍처 정의 작업(1R21)	54
표준 지침 수립 작업(1R22)	77
유스케이스 모델링 작업(1R31)	81
연동 소요 분석 작업(1R32)	90
사용자 인터페이스 정의 작업(1R33)	104
클래스 모델링 작업(1R34)	112
자료 구축 계획 작업(1R41)	119
테스트 계획 작업(1R51)	127
 설계 단계(2D)	 136
컴포넌트 식별 작업(2D11)	139
인터페이스 상호작용 명세 작업(2D12)	148
컴포넌트 명세 작업(2D13)	152
사용자 인터페이스 명세 작업(2D14)	161
데이터 모델링 작업(2D15)	170
컴포넌트 구현 설계 작업(2D21)	175
사용자 인터페이스 구현 설계 작업(2D22)	189
데이터베이스 설계 작업(2D23)	193
연동 설계 작업(2D24)	217
자료 구축 설계 작업(2D25)	236
시스템 보안 설계 작업(2D26)	245
시스템 설치 계획 작업(2D31)	250

구현 및 테스트 단계(3T)	260
데이터베이스 구축 작업(3T11)	263
컴포넌트 구현 작업(3T12)	267
사용자 인터페이스 구현 작업(3T13)	272
단위 테스트 준비 작업(3T21)	276
단위 테스트 수행 작업(3T22)	284
소프트웨어 통합 테스트 준비 작업(3T31)	289
소프트웨어 통합 및 테스트 수행 작업(3T32)	296
시스템 통합 테스트 준비 작업(3T41)	301
시스템 통합 및 테스트 수행 작업(3T42)	309
사용자 지침서 작성 작업(3T51)	314
운용자 지침서 작성 작업(3T52)	319
 인도 단계(4S)	 326
시스템 설치 실시 작업(4S11)	327
인수 지원 실시(4S21)	335

분석 단계(1R)

개요

분석 단계는 개발 대상 시스템의 요구사항과 이를 만족시키는 시스템 아키텍처를 정의하고 요구사항을 시스템 관점에서 분석해 개발 대상 시스템이 제공해야 하는 기능, 시스템 운용에 필요한 자료 구축 및 테스트 계획을 수립하는 것을 목적으로 한다. 분석 단계는 요구사항 식별, 아키텍처 정의, 요구사항 분석, 자료 구축 준비, 테스트 준비 활동으로 구성된다.

활동/작업 설명

1. 요구사항 식별(1R1)

요구사항 식별 활동은 개발 대상 시스템이 속한 도메인을 정의하고 개발 대상 시스템의 배경 및 관련 업무에 대한 명확한 이해를 토대로 상세한 요구사항을 정의한다.

작업ID	작업명	작업내용
1R11	도메인 모델링	시스템과 관련된 훈령, 지침과 더불어 현행 업무 관련문서, 개념연구 문서 등을 통해 도메인의 범위를 정하고 개발 시스템과 상호 작용하는 외부 객체들 사이의 관계를 나타내며 프로젝트 용어를 정의한다.
1R12	현행 시스템 분석	개발 대상 시스템에 대한 이해를 돕기 위해 현재 운용 중인 시스템의 구성과 환경 등을 분석한다. 이를 통해 현행 시스템의 문제점을 파악하고 해결방안을 도출한다.
1R13	비즈니스 모델링	개발 대상 시스템이 지원해야 할 비즈니스 프로세스, 수행자 및 관련 정보 등을 식별하고 그들 간의 관계에 대해서 정의한다.
1R14	요구사항 정의	수집한 자료 및 선행 작업 결과를 바탕으로 개발 대상 시스템의 목표와 상위 요구사항을 정의한 후, 사용자와의 면담, 설문, 자체 식별 등을 통해 개발 대상 시스템의 상세한 요구사항을 정의한다.

분석 단계(1R)

2. 아키텍처 정의(1R2)

개발 대상 시스템의 운용 환경과 하드웨어, 소프트웨어 및 네트워크 구조 등을 정의하고, 프로그램 개발을 위한 소프트웨어 아키텍처를 선정하며 프로젝트 진행에 필요한 표준을 정의한다.

작업ID	작업명	작업내용
1R21	시스템 아키텍처 정의	요구사항을 충족할 수 있도록 개발 대상 시스템의 소프트웨어 아키텍처와 하드웨어 및 네트워크 구조 등을 정의한다.
1R22	표준 지침 수립	시스템 개발 시 준수해야 할 표준 및 지침을 정의하여 개발자들 간의 이해와 공유를 도모하고 시스템이 체계적으로 개발될 수 있도록 한다.

3. 요구사항 분석(1R3)

정의된 요구사항을 바탕으로 개발 대상 시스템이 제공해야 할 기능을 분석한다.

작업ID	작업명	작업내용
1R31	유스케이스 모델링	개발 대상 시스템과 상호작용하는 액터와 시스템의 기능인 유스케이스를 식별하고 이들 간의 관계를 정의하여 요구사항을 구체화한다.
1R32	연동 소요 분석	개발 대상 시스템과 타 시스템간의 연동 소요가 존재할 경우 연동을 위한 환경, 기술 및 연동 항목 등을 식별하고 실현 방안을 도출한다.
1R33	사용자 인터페이스 정의	사용자 인터페이스 흐름을 도출하고 화면, 보고서에 대한 레이아웃 및 구성항목을 정의한다.
1R34	클래스 모델링	개발 대상 시스템을 구성하는 클래스를 식별하고 클래스 속성 및 오퍼레이션을 정의한다. 식별된 클래스 간의 상호작용 관계를 분석해 클래스를 정제하면서 클래스간의 관계를 정의한다.

분석 단계(1R)

4. 자료 구축 준비(1R4)

개발 대상 시스템을 운용하기 위해 필요한 자료 전환 및 초기 자료 구축 대상을 식별하고, 자료 전환 및 초기 자료 구축 계획을 수립한다.

작업ID	작업명	작업내용
1R41	자료 구축 계획	구축 대상 자료를 선정하고, 구축 방법 및 전략과 세부 일정계획 등을 수립한다.

5. 테스트 준비(1R5)

시스템 개발 결과를 테스트하기 위해 테스트의 종류, 일정, 수행 기준 및 방법을 정의한다.

작업ID	작업명	작업내용
1R51	테스트 계획	수행해야 할 테스트의 종류 및 대상을 식별하고 테스트 일정, 수행 기준 및 방법, 환경 등 전반적인 테스트 계획을 수립한다. 테스트의 종류는 방법론에서 제시된 단위 테스트, 소프트웨어 통합 테스트, 시스템 통합 테스트 외에 필요하거나 요구되는 테스트를 추가할 수 있다.

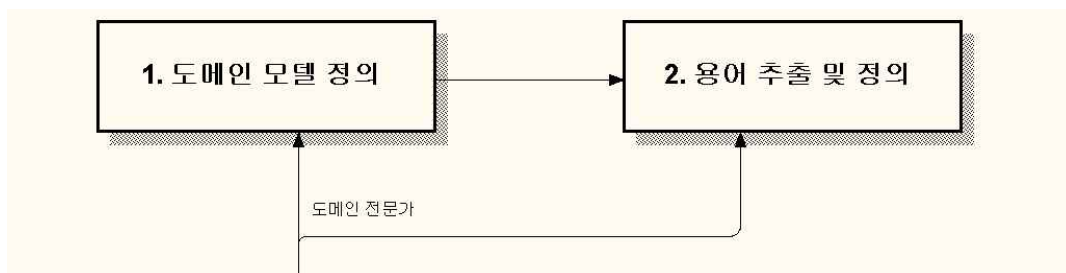
도메인 모델링 작업(1R11)

개요

개발 대상 시스템의 도메인을 정의하고 시스템과 시스템 외부 요소들 간의 관계를 통해 개발 대상 시스템을 이해하는 작업이다. 또한, 도메인을 중심으로 개발 과정에서 통용되는 용어와 약어를 정의해 시스템 개발 및 문서작성, 의사소통 시 기준으로 활용한다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 훈령 및 지침 • 국방정보기술아키텍처 • 개념연구 문서 • 현행 관련 문서
산출물	<ul style="list-style-type: none"> • 도메인 정의서, 용어집
참여자	<ul style="list-style-type: none"> • 도메인 전문가 <p>도메인에 대한 전문 지식을 갖춘 자로서 개발 대상 시스템이 속해있는 도메인의 범위를 정의하고 시스템과 시스템 외부 요소들 간의 관계를 정의한다.</p>
관련기법	<ul style="list-style-type: none"> • UML, DFD, IDEF0
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 도메인 모델 정의

개발 대상 시스템이 속한 도메인을 정의하고 다른 도메인 또는 개발 대상 시스템과 상호작용하는 시스템 외부 객체와의 관계를 식별하여 도메인 모델로 나타낸다. 시스템 외부 객체는 타 시스템이나 운용 환경 등을 의미한다.

■ 도메인 모델링 작업(1R11)

- 도메인 정의
개발 대상 시스템과 관련된 자료(훈령 및 지침, 국방정보기술아키텍처, 개념연구 문서 등)를 기반으로 개발 대상 시스템이 속한 도메인을 정의한다.
- 개발 대상 시스템의 상호작용 내역 식별
개발 대상 시스템과 상호작용하는 시스템 외부 객체를 식별하고 그에 따른 제어 흐름이나 데이터 흐름 등 상호작용 내역을 식별한다.
- 도메인내 공통성 분석
도메인의 범위와 특징(업무, 기능, 데이터 등)을 기반으로 도메인 내의 시스템 간 공통성과 차별성을 분석해 개발 대상 시스템 또는 새로운 시스템을 개발할 때 재사용 될 수 있는 부분을 식별하고 시스템간의 상호작용 관점의 연관성을 식별한다.

2. 용어 추출 및 정의

도메인을 중심으로 용어를 추출하여 표준 용어를 정의한다.

- 표준 용어 정의
도메인에서 사용하는 용어를 추출해 표준 용어를 선정하고 각 표준 용어에 대한 설명, 동의어를 정의한다. 초기에는 도메인을 중심으로 용어를 정의하지만, 프로젝트를 수행하면서 발견되는 용어를 계속적으로 추가하여 정의한다.

■ 작업 수행시 고려사항

■ 도메인 모델링 작업의 수행 여부

- 도메인 모델 정의는 개발 대상 시스템의 도메인과 상호작용하는 시스템 외부 객체가 이미 명확히 식별되어 있고 국방정보기술아키텍처 등에서 작성한 도메인 모델 정의 관련 산출물이 존재하며 세부 내용에 차이가 없는 경우 해당 산출물로 작업 수행을 대체할 수 있다.

■ 도메인 모델 정의시 고려사항

- 도메인은 다른 도메인과 범위가 중첩되지 않도록 식별해야 하며 업무, 기능, 데이터 등에서 특징이 있어야 한다.
- 도메인 모델은 개발자가 시스템의 내/외부 환경을 이해해 사용자와의 의사소통을 원활히 하고 사용자 요구사항을 명확히 이해할 수 있도록 하기 위한 것으로 도메인 및 개발 대상 시스템이 구체적으로 표현되도록 정의한다.
- 도메인 모델의 표현은 UML(Unified Modeling Language), DFD(Data Flow Diagram), IDEF0(Integration DEFinition Method) 등 다양한 표기법을 사용해 작성할 수 있다.

도메인 모델링 작업(1R11)

■ 용어 정의시 고려사항

- 사용자측에 이미 정의된 용어집(업무 수행을 위한 지침 또는 국방정보기술아키텍처 등)이 존재한다면 이를 우선적으로 사용하고, 필요 시 사용자와의 조정 협의를 통해 수정한다. 정의된 표준 용어는 교육, 발표회, 홍보물 등을 이용해 이해관계자 모두가 인지하도록 한다.
- 용어는 한글 용어를 중심으로 기술하고(용어가 한글과 영문 용어를 함께 사용하는 경우 포함) 영문 용어가 외래어이거나 일반적으로 영문 용어만을 사용하는 경우에는 영문 용어를 중심으로 기술한다.
- 약어를 주로 사용하는 용어는 약어를 중심으로 한글과 영문 용어를 기술한다.
- 정의된 표준 용어를 기준으로 현재 사용되고 있는 유사 용어가 존재하는 경우 식별하여 기술한다.

■ 도메인 모델링 결과와 이후 작업과의 관계

- 정의된 도메인, 시스템 외부 객체를 현행 시스템 분석(1R12)에서 분석 대상 식별 기준으로 활용할 수 있다.
- 개발 대상 시스템과 상호작용하는 시스템 외부 객체는 비즈니스 모델링(1R13)에서 비즈니스 액터로 식별되거나 요구사항 정의(1R14)시 비기능적 요구사항으로 식별될 수 있다.
- 도메인내 공통성은 향후 시스템 개발 시 재사용 가능한 부분으로 유도되어 유스케이스 모델링(1R31)에서 다른 유스케이스에 포함/확장 되는 유스케이스로 식별되거나 별도의 공통 유스케이스로 식별될 수 있다.

도메인 모델링 작업(1R11)

[서식] 도메인 정의서(1R11a)

1. 도메인 모델

1.1. 개요

1.2. 도메인 구성도

1.3. 공통성 정의

도메인 모델링 작업(1R11)

[항목 설명] 도메인 정의서(1R11a)

1. 도메인 모델

1.1. 개요

개발 대상 시스템이 속한 도메인을 정의하고, 해당 도메인에 대한 설명을 기술한다. 도메인에 대한 설명은 도메인의 범위, 특징(업무, 기능, 데이터 등)에 초점을 두어 기술한다.

1.2. 도메인 구성도

개발 대상 시스템이 다른 도메인과 어떤 관계가 있는지, 시스템 외부 객체(타 시스템이나 타 운용환경 등)와의 상호작용 관계가 어떻게 이루어지는지를 제어 흐름 또는 데이터 흐름 등과 함께 도식적으로 표현한다.

1.3. 공통성 정의

같은 도메인에 있는 여러 시스템들을 분석해서 도메인 내의 공통적인 기능들을 찾아 정의한다.

도메인 모델링 작업(1R11)

[서식] 용어집(1R11b)

1. 용어 정의

용어 명칭	용어 정의	한글/영문명		약어		용어 적용 범위	기타 사항
		한글명	영문명	한글약어	영문약어		

도메인 모델링 작업(1R11)

■ [항목 설명] 용어집(1R11b)

1. 용어 정의*

- 용어 명칭
용어를 정의하기 위한 명칭을 기재한다. 한글 용어일 경우 한글명 항목을, 영문 용어일 경우 영문명 항목은 작성하지 않는다. 용어가 약어일 경우 한글, 영문 약어를 작성하지 않는다.
- 용어 정의
용어에 대한 상세한 정의를 기술한다.
- 한글명
용어에 대한 한글 명칭을 기재한다.
- 영문명
용어에 대한 영문 명칭을 기재한다.
- 한글 약어
용어에 대한 한글 약어를 기재한다.
- 영문 약어
용어에 대한 영문 약어를 기재한다.
- 용어 적용 범위
용어가 적용되는 범위에 대한 구분을 기재한다.

구분	전군 / 단위사업
----	-----------

- 기타 사항
유사 용어 및 용어의 이해와 관리를 위해 필요한 부가적인 정보를 기술한다.

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

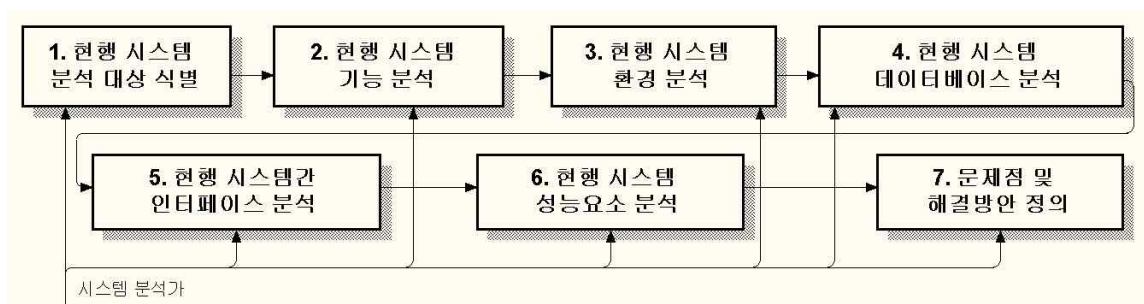
■ **현행 시스템 분석 작업(1R12)**

■ **개 요**

개발 대상 시스템에 대한 이해를 돕고 개선 소요를 식별해 요구사항으로 반영하기 위해서 현행 시스템의 환경, 기능 및 구성 등을 분석하는 작업이다. 또한, 현행 시스템의 운용 환경을 이해하고 문제점을 파악하여 그에 대한 해결 방안을 도출한다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 계약 관련 문서(제안요청서, 제안서, 기술 협상 문서, 계약서 등) • 훈령 및 지침 • 국방정보기술아키텍처 • 개념연구 문서 • 현행 관련 문서 • 도메인 정의서
산출물	<ul style="list-style-type: none"> • 현행 시스템 분석서
참여자	<ul style="list-style-type: none"> • 시스템 분석가 <p>현재 운영 중인 시스템을 이해하여 문제점을 파악하고 해결방안을 도출한다.</p>
관련기법	<ul style="list-style-type: none"> • 브라운 페이퍼 분석
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2

■ **절차 흐름도**



■ **현행 시스템 분석 작업(1R12)**

■ **절차 설명**

1. 현행 시스템 분석 대상 식별

개발 대상 시스템의 현행 시스템을 식별한다.

- **현행 시스템 분석 대상 식별**

개발 대상 시스템이 속한 도메인 내에서 요구되는 기능, 관련 업무, 데이터, 기술 관점에서 유사한 시스템을 분석 대상으로 우선 식별하고 타 도메인 또는 개발 대상 시스템과 상호작용하는 시스템을 기준으로 현행 시스템 분석 대상을 식별한다.

2. 현행 시스템 기능 분석

현행 시스템이 제공하는 기능을 분석한다.

- **현행 시스템 기능 분석**

개발 대상 시스템에 요구되는 기능에 초점을 두어 현행 시스템이 제공하는 세부 기능과 기능간의 관계 및 주요 특성 등을 분석하여 기술하고 전체적인 시스템 기능 구조를 도식적으로 표현한다.

3. 현행 시스템 환경 분석

현행 시스템이 운용되는 하드웨어, 네트워크 등 기반 환경을 분석한다.

- **시스템 노드 식별**

현행 시스템이 운용되는 물리적 위치(시설 등), 논리적 위치(조직, 무기체계 등) 등의 시스템 노드를 식별한다.

- **기반 환경 식별**

현행 시스템이 운용되는 하드웨어, 패키지/시스템 소프트웨어와 네트워크 및 통신장비 등 기반 환경을 식별하고 이들 간의 관계를 정의한다. 또한, 하드웨어 및 네트워크의 전체 구성에 대해 도식적으로 표현한다.

- **추가 사항 식별**

장소, 항온항습기, 전력, 라이선스 등의 제약사항이나 전제조건과 같은 추가 사항에 관한 분석이 필요할 경우 해당 내용을 조사, 분석한다.

■ **현행 시스템 분석 작업(1R12)**

4. 현행 시스템 데이터베이스 분석

현행 시스템의 데이터 구조를 이해하기 위해 데이터베이스를 분석한다.

- 현행 시스템의 데이터베이스, 테이블 및 인덱스 정보 분석
- 현행 시스템의 코드 체계와 코드 값 분석

5. 현행 시스템간 인터페이스 분석

현행 시스템의 시스템간 인터페이스를 식별한다.

- 현행 시스템간 인터페이스 식별
현행 시스템을 기준으로 연관된 시스템과의 인터페이스 관계를 식별한다.
- 인터페이스 내역 분석
각각의 인터페이스에서 교환하는 데이터에 대한 설명, 주기, 통신매체 및 크기 등 인터페이스 내역을 분석한다.

6. 현행 시스템 성능요소 분석

현행 시스템이 성능상의 문제점이 존재하거나 개선이 필요한 경우 시스템 구성 요소들에 대한 성능 테스트 요소 또는 항목을 선정하고 테스트를 진행한다.

- 분석할 현행 시스템의 성능요소 식별
- 개발 대상 시스템에서 요구되거나 필요로 하는 성능 사항 분석

7. 문제점 및 해결방안 정의

분석된 현행 시스템 정보를 기반으로 현 시스템의 업무 지원 부족 또는 업무와 현 시스템 기능간의 불일치 등과 같은 현행 시스템의 문제점을 식별하고 이에 대한 해결 방안을 정의한다.

- 현행 시스템 문제점 식별 및 분석
현행 시스템 분석을 통해 확인된 문제점을 식별하고 문제점이 발생하는 원인을 분석한다.
- 해결방안 정의
문제점별로 해결방안을 정의한다.

■ **현행 시스템 분석 작업(1R12)**

■ **작업 수행시 고려사항**

■ **현행 시스템 분석 작업의 수행 여부**

- 현행 시스템 분석 작업은 현재 운용 중인 현행 시스템이 존재하거나 개발 대상 시스템과 유사한 시스템이 존재하여야 수행이 가능하다. 따라서 분석 대상의 선정이 중요하므로 개발자가 현행 시스템 대상을 식별한 후 사용자와의 협의를 통해 수행 여부를 결정한다.
- 사전에 획득한 현행 관련 문서 또는 현행 시스템의 개발 산출물이 존재하고, 그것을 통해 충분한 분석이 가능한 경우 사용자와의 협의를 통해 관련 문서로 대체할 수 있다.

■ **현행 시스템 분석 대상 식별시 고려사항**

- 재개발 또는 성능개선과 같이 현행 시스템이 명확히 존재하는 경우를 제외한 다면, 현행 시스템의 대상을 선정하는데 있어 사용자와 개발자간의 관점의 차이가 존재할 수 있다. 일반적으로, 현행 시스템은 개발 대상 시스템과 동일한 목적을 가지고, 동일한 도메인에 속하며, 동일한 기능을 제공하는, 현재 운용 중에 있는 시스템이 해당된다.
- 현행 시스템에 해당되는 시스템이 없는 경우 업무 프로세스나 데이터 유형이 유사한 시스템을 분석 대상으로 식별할 수 있다. 현행 시스템 분석 대상을 식별한 후에는 해당 시스템에 영향을 미치는 규정 및 지침을 함께 식별한다.

■ **현행 시스템 분석시 고려사항**

- 현행 시스템 분석 작업에는 해당 시스템을 운용/관리하는 부서의 협조가 중요하다. 수집한 현행 시스템 정보가 불충분할 경우, 개발자가 직접 현행 시스템을 분석하여 정보를 획득하고, 필요시 면담, 설문 등을 통해 시스템 서비스 수준과 관리상 문제점을 파악해야 한다.
- 현행 시스템 분석을 통하여 개발 대상 시스템에서 개선하거나 변경하고자 하는 것에 대하여 더 많이 이해할 수 있으며 현행 시스템의 숨겨진 문제점을 식별할 수 있다.
- 개발 대상 시스템의 운용을 위해 초기 자료 구축 또는 기존 운영 자료의 전환이 필요한 경우 별도의 자료 수집 절차를 수행하고 수집된 자료를 관리해야 한다.
- 개발 대상 시스템이 지원하는 업무가 근무 종료 시점, 월/분기 단위의 현황보고, 정기적인 데이터 수집 또는 계획 수립 등과 같이 특정 시점에서 업무량이 증가하여 성능적인 측면이 고려되어야 한다면 다음의 항목에 초점을 두어 성능 요소를 분석한다.

■ **현행 시스템 분석 작업(1R12)**

- 주요 업무의 시간대별 동시 사용자(Active Users) 수 산정
 - 시간대, 주기별 트랜잭션 처리량
 - 네트워크 트래픽 형태
 - 서버 자원 사용량
 - 회선 사용량
- 현행 시스템을 분석하는데 너무 많은 시간을 소모하지 않도록 주의해야 하며 새로운 시스템의 요구 사항을 위한 모델을 개발하는 데에 더 많은 노력을 기울이는 것이 중요하다.

■ **현행 시스템 분석서 서식 적용**

- 현행 시스템 분석서의 서식은 국방아키텍처프레임워크(MND-AF) v1.2에서 제시된 산출물의 항목들과 많은 부분이 일치한다. 이는 국방정보기술아키텍처의 AS-IS 아키텍처를 활용하기 위한 것으로 현행 시스템이 국방정보기술아키텍처로 구축된 경우 체계관점 산출물들을 참조하여 현행 시스템 분석서를 작성한다.
- 현행 시스템의 테이블과 코드 등에 관련된 자료가 별도로 존재하고 그 자료를 통해 충분한 데이터 구조의 분석이 가능한 경우 현행 시스템 분석서 서식의 데이터베이스 구성 항목에서 테이블, 코드, 세부코드의 작성을 대체할 수 있다.
- 현행 시스템과 관련한 기존 산출물이 미흡할 경우, 제시된 현행 시스템 분석서 서식을 모두 적용하지 않아도 무방하다. 다만, 본 작업은 현행 시스템 분석을 통해 운용 과정을 이해하고 현행 시스템의 문제점을 파악하며 그에 대한 해결 방안을 도출하기 위한 것으로 목적에 맞게 산출물을 재구성하여 작성한다.
- 일반적으로 하드웨어와 네트워크 구성은 분리해 작성하기 곤란하며 하드웨어, 네트워크, 소프트웨어 등 기반 환경을 통합하여 도식화가 가능하다면 하나의 구성도로 작성한다.

■ **현행 시스템 분석 작업(1R12)**

■ [서식] **현행 시스템 분석서(1R12a)**

1. (현행 시스템명)

1.1. 관련 지침

지침명	설명	분야	제정년월	최근 개정년월	제정기관

1.2. 현행 시스템 개요

시스템명		상위 시스템명			
설명					
약어		버전		상태	
사용환경		분산구조		개발언어	
상호운용성 도달가능수준		도메인명			
운용조직		사용조직			

■ **현행 시스템 분석 작업(1R12)**

1.3. 시스템 기능

기능명	기능 설명	기능 수준	개발 형태	상위 기능명	관련 테이블

1.4. 하드웨어 구성

1.4.1. 하드웨어 구성도

1.4.2. 하드웨어

하드웨어명	설명	장비유형	용도	수량	Port수	Disk 용량	메모리	CPU 종류	CPU 수량	CPU 속도	제조사	OS

1.5. 네트워크 및 통신장비 구성

1.5.1. 네트워크 구성도

■ **현행 시스템 분석 작업(1R12)**

1.5.2. 시스템 노드

시스템 노드명	설 명

1.5.3. 네트워크

네트워크명	설 명	유형	속도	용도	보안등급	관련 시스템 노드

1.5.4. 랜

랜명	설 명	속도	관련 통신장비	관련 하드웨어

■ **현행 시스템 분석 작업(1R12)**

1.5.5. 통신장비

통신장비명	설명	유형	수량	모델명	제조사	관련 하드웨어	시스템 노드

1.6. 소프트웨어 구성

1.6.1. 패키지 소프트웨어

소프트웨어명	설 명

1.6.2. 시스템 소프트웨어

하드웨어명	유형	소프트웨어명	설명	버전	제조사

■ **현행 시스템 분석 작업(1R12)**

1.7. 데이터베이스 구성

1.7.1. 데이터베이스

데이터베이스명	설명

1.7.2. 테이블

테이블명				설명			
데이터 건수				데이터베이스명			
인덱스명				인덱스 설명			
컬럼명	설명		속성키 구분	데이터 타입		길이	코드 여부

1.7.3. 코드

코드ID	코드명	코드 영문명	코드 영문약어명	설명	데이터타입	데이터길이	표준 코드 여부	비고

■ **현행 시스템 분석 작업(1R12)**

1.7.4. 세부코드

코드ID	코드명	코드값	코드값명	상위 코드명	상위 코드값	상위 코드값명

1.8. 타 시스템간 인터페이스

1.8.1. 인터페이스 구성도

1.8.2. 인터페이스

인터페이스명	설명	출발 시스템명	도착 시스템명

■ **현행 시스템 분석 작업(1R12)**

1.8.3. 인터페이스 데이터

인터페이스 명	인터페이스 데이터명	설명	상호운용성 달성가능수준	주기	처리량	적시성	보안 등급	통신매체 유형	크기	단위	포맷 유형	트리거 이벤트

2. 성능요소

3. 문제점 및 해결방안

문제점명	설명	해결방안

■ **현행 시스템 분석 작업(1R12)**

■ **[항목 설명] 현행 시스템 분석서(1R12a)**

1. (현행 시스템명)

현행 시스템별로 작성한다. 구성도 등을 이용해 각 항목의 표현이 가능할 경우 해당 항목은 작성하지 않아도 무방하다.

1.1. 관련 지침*

- 지침명
지침을 대표하는 명칭을 기재한다.
- 설명
지침에 대한 설명을 기술한다.
- 분야
지침이 속한 대표적인 분야를 기재한다.

구분	시스템관리 / 보안 / 구매 / 운영관리 / 기타(직접기재)
-----------	-----------------------------------

- 제정년월
지침이 최초로 제정된 년/월을 기재한다.
- 최근 개정년월
지침이 최근에 개정된 년/월을 기재한다.
- 제정기관
지침을 제정한 기관의 명칭을 기재한다.(예 : 국방부, 방위사업청)

1.2. 현행 시스템 개요*

- 시스템명
시스템을 대표하는 명칭을 기재한다.
- 상위 시스템명
시스템을 포함하는 상위 시스템이 있을 경우 이를 대표하는 명칭을 기재한다.
- 설명
시스템에 대한 설명을 기술한다.
- 약어
시스템을 나타내는 약어로 한글약어 또는 영문약어로 표현할 수 있으며, 일반적으로 영문약어로 기재한다. (예 : KJCCS, MIMS 등)
- 버전
시스템의 현 버전 정보를 기재한다. (예 : v1.1, v1.2)

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

현행 시스템 분석 작업(1R12)

- 상태

생애주기 측면에서 시스템의 상태를 구분하여 기재한다.

구분	개발완료 및 운용 중 / 개발 중 또는 운용예정 / 개발예정 / 개발미정 / 폐기예정 / 폐기
----	--

- 사용환경

시스템이 불특정 다수에 의해 사용되는지 또는 내부에서 사용되는 시스템인지 등을 식별하기 위한 속성을 구분하여 기재한다.

구분	인터넷 / 인트라넷 / 엑스트라넷 / 직접기재 / 미정의 / 기타(직접기재)
----	--

- 분산구조

시스템이 운용될 환경의 분산 구조를 기재한다.

구분	호스트기반 / 클라이언트/서버기반/ 웹기반 / 홈페이지 / 기타(직접기재)
----	---

- 개발언어

시스템 개발에 사용되어진 프로그래밍 언어를 기술한다.

- 상호운용성 도달 가능 수준

시스템이 달성되거나 달성가능한 상호운용성 수준(Level of Information System Interoperability)을 기재한다.

구분	0 : Level 0-하위 수준이 없는 격리된 수준 / 1 : Level 1-하위 수준이 없는 연결된 수준 / 1a : Level 1-연결된 수준, 하위수준 a / 1b : Level 1-연결된 수준, 하위수준 b / 1c : Level 1-연결된 수준, 하위수준 c / 1d : Level 1-연결된 수준, 하위수준 d / 2 : Level 2-하위 수준이 없는 기능적인 수준 / 2a : Level 2-기능적인 수준, 하위수준 a / 2b : Level 2-기능적인 수준, 하위수준 b / 2c : Level 2-기능적인 수준, 하위수준 c / 3 : Level 3-하위수준이 없는 도메인 수준 / 3a : Level 3-도메인 수준, 하위수준 a / 3b : Level 3-도메인 수준, 하위수준 b / 3c : Level 3-도메인 수준, 하위수준 c / 4 : Level 4-하위 수준이 없는 엔터프라이즈 수준 / 4a : Level 4-엔터프라이즈 수준, 하위수준 a / 4b : Level 4-엔터프라이즈 수준, 하위수준 b / 4c : Level 4-엔터프라이즈 수준, 하위수준 c / 기타(직접기재)
----	--

■ **현행 시스템 분석 작업(1R12)**

- 도메인명
논리적으로 유사한 시스템의 그룹인 시스템 도메인의 명칭을 기재한다.
- 운용조직
현행 시스템을 운용하는 조직명을 기재한다.
- 사용조직
현행 시스템을 사용하는 조직명을 기재한다.

1.3. 시스템 기능*

- 기능명
시스템 기능을 식별하기 위한 고유한 명칭을 기재한다.
- 기능 설명
시스템 기능에 대한 설명을 기술한다.
- 기능 수준
시스템 기능간의 상하관계를 나타내며, 최상위 수준을 “1”로 표시하고 하위 수준은 “점_숫자”의 형태로 기재한다. (예 : 1.1, 1.1.21, 1.1.21.1 등)
- 개발 형태
시스템 기능 개발의 형태를 기재한다.

구분	제품활용 / 커스터마이징 / 자체개발 / 기타(직접기재)
----	---------------------------------

- 상위 기능명
상위 기능을 식별하기 위한 고유한 명칭을 기재한다.
- 관련 테이블
해당 시스템 기능과 연관된 테이블명을 기재한다.

1.4. 하드웨어 구성

1.4.1. 하드웨어 구성도

시스템내의 하드웨어와 각각의 연결 정보를 세부적으로 표현한 개념도로, 체계의 물리적 하드웨어 등에 대한 구성을 전체적으로 표현한다. 1.5.1 네트워크 구성도와 통합해 작성할 수 있다.

1.4.2. 하드웨어*

- 하드웨어명
시스템을 구성하는 하드웨어에 대한 명칭을 기재한다.
- 설명
하드웨어에 대한 구체적인 설명을 기술한다.

현행 시스템 분석 작업(1R12)

- 장비유형

하드웨어 장비 유형 정보에 대해 구분하여 기재한다.

구분	개인용 PC / PC급 서버 / 중형서버 / 대형서버 / 디스크어레이 / 백업장치 / 기타(직접기재)
----	--

- 용도

하드웨어에 대한 용도를 구분하여 기재한다.

구분	내부업무처리용 (MIS, 문서처리 등) / 시스템 내 응용시스템 서버 / 백업용 서버 / 파일서버 / 보안용 / 기타(직접 기재)
----	--

- 수량

하드웨어 수량에 대해 기재한다.

- Port수

하드웨어가 보유하고 있는 총 포트 수를 기재한다.

- Disk용량

하드웨어가 보유하고 있는 디스크 총 용량을 기재한다.

- 메모리

총 메모리 용량을 기재한다.

- CPU종류

하드웨어 CPU 종류를 기재한다.

- CPU수량

하드웨어 CPU 총 개수를 기재한다.

- CPU속도

하드웨어 CPU 속도를 기재한다.

- 제조사

하드웨어에 대한 제조사를 기재한다.

- OS

하드웨어에 탑재된 운영체제를 기재한다.

1.5. 네트워크 및 통신장비 구성

1.5.1. 네트워크 구성도

네트워크 구성을 가시적으로 확인할 수 있도록 네트워크, 통신장비, 랜, 하드웨어로 구성되는 개념도를 작성하되, 네트워크가 여러 개일 경우 필요시 각각 작성할 수 있다. 네트워크 구성도는 시스템 노드별로 작성한다.

■ 현행 시스템 분석 작업(1R12)

1.5.2. 시스템 노드*

- 시스템 노드명
시스템이 운용되는 실질적 노드를 기재한다. (예 : ○○부대 전산실)
- 설명
시스템 노드에 대한 설명을 기술한다.

1.5.3. 네트워크*

- 네트워크명
시스템이 운용되는 네트워크 명칭을 기재한다.
- 설명
목적, 규모 등 네트워크에 대한 상세한 설명을 기술한다.
- 유형
네트워크를 구분하기 위한 유형을 기재한다.

구분	LAN / WAN / VPN / ATM / BcN / Internet / Intranet / 백업회선 / 재해복구회선 / 기타(직접 기재)
----	---

- 속도
네트워크의 속도를 기재한다.
- 용도
네트워크의 사용 용도를 기술한다.
- 보안등급
'보안업무준령'에 정의된 보안등급을 기재한다.

구분	평문 / 대외비 / III급 / II급 / I급 / SI / 기타(직접 기재)
----	---

- 관련 시스템 노드
네트워크가 운용되는 실질적 노드를 기재한다. (예 : ○○부대 전산실)

1.5.4. 랜*

- 랜명
시스템 노드 내에서 통신장비와 하드웨어간 또는 하드웨어간을 연결하는 물리적인 랜의 명칭을 기재한다.
- 설명
랜에 대한 구체적인 설명을 기술한다. (예 : 토폴로지 유형, 규모 등)
- 속도
내부 속도를 기재한다. (예 : 10MBps, 100MBps 등)

■ 현행 시스템 분석 작업(1R12)

- 관련 통신장비
랜에 연결된 통신장비명을 기재한다.
- 관련 하드웨어
랜에 연결된 하드웨어명을 기재한다.

1.5.5. 통신장비*

- 통신장비명
데이터 혹은 정보의 전송 기능을 가지고 있는 통신장비의 명칭을 기술한다. 예로 네트워크 스위치와 라우터, 위성 등을 포함한다.
- 설명
통신장비의 통신기능에 대한 간단한 설명을 기술한다.
- 유형
통신장비의 유형을 구분하여 기재한다.

구분	방화벽 / 라우터 / 보안장비 / 허브/ 스위치 / 모뎀 / 게이트웨이 / VPN / 기타(직접 기재)
----	---

- 수량
통신장비 수량에 대해 기재한다.
- 모델명
통신장비의 제조 모델명을 기재한다.
- 제조사
통신장비에 대한 제조사를 기재한다.
- 관련 하드웨어
통신장비와 통신하는 하드웨어를 기재한다.
- 시스템 노드
통신장비가 운용되는 실질적 노드를 기재한다. (예 : ○○부대 전산실)

1.6. 소프트웨어 구성

1.6.1. 패키지 소프트웨어*

- 소프트웨어명
패키지로 판매되는 솔루션 형태의 소프트웨어 집합의 명칭을 기재한다.
(예 : ERP, NMS, DW 등)
- 설명
패키지 소프트웨어에 대한 구체적인 설명을 기술한다.

■ 현행 시스템 분석 작업(1R12)

1.6.2. 시스템 소프트웨어*

- 하드웨어명

시스템 소프트웨어가 탑재되는 하드웨어명을 기재한다.

- 유형

시스템 소프트웨어를 구분하기 위한 유형을 기재한다.

구분	운영체제 / 미들웨어 / DBMS / 문서편집기 / 스프레드시트 / 모델링툴 / 그래픽툴 / 메일 / 백신 / 보안 / 기타(직접기재)
----	---

- 소프트웨어명

시스템 소프트웨어 명칭을 기재한다.

- 설명

시스템 소프트웨어에 대한 구체적인 설명을 기술한다.

- 버전

시스템 소프트웨어에 대한 릴리즈를 포함한 버전을 기재한다.

- 제조사

시스템 소프트웨어에 대한 제조사를 기재한다.

1.7. 데이터베이스 구성

1.7.1. 데이터베이스

- 데이터베이스명

데이터의 최상의 집합인 데이터베이스 명칭을 기재한다.

- 설명

데이터베이스에 대한 상세한 설명을 작성한다. 현행 시스템 분석 수준에 따라 테이블이나 코드에 대한 상세 분석이 필요하지 않은 경우 데이터베이스 설명에 테이블과 코드에 대해 추가 설명한다.

1.7.2. 테이블

- 테이블명

테이블의 명칭을 기재한다.

- 설명

테이블에 대한 간단한 설명을 기술한다.

- 데이터 건수

테이블에 저장되어 있는 데이터의 건수를 기재한다.

- 데이터베이스명

테이블이 속한 데이터베이스 명칭을 기재한다.

■ 현행 시스템 분석 작업(1R12)

- 인덱스명
테이블에 표현된 인덱스 명칭을 기재한다.
- 인덱스 설명
인덱스의 기능에 대하여 간단한 설명을 기술한다.
- 컬럼명
각 속성에 대응하는 컬럼의 이름을 기재한다.
- 설명
컬럼에 대한 간단한 설명을 기술한다.
- 속성키 구분
해당 컬럼이 키로 사용되는지를 체크한다.
- 데이터 타입
각 컬럼의 데이터 유형을 기재한다.
- 길이
각 컬럼의 길이를 표시한다.
- 코드 여부
해당 컬럼이 코드일 경우, 정확한 코드 명칭을 기술한다.

1.7.3. 코드**

- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재한다.
- 코드명
코드를 대표하는 명칭을 기재한다.
- 코드 영문명
코드의 영문명을 기재한다.
- 코드 영문약어명
코드의 영문명에 대한 약어명을 기재한다.
- 설명
코드에 대한 개략적인 설명을 기술한다.
- 데이터 타입
코드의 논리적 데이터 타입(문자형, 숫자형, 날짜형 등)을 나타내며 MDR(Meta Data Repository)에서 제시된 타입을 준용하여 지정한다.
- 데이터 길이
코드의 최대 데이터 길이를 기재한다.

** 데이터 표준화 지침 유관 항목

■ **현행 시스템 분석 작업(1R12)**

- 표준 코드 여부
해당 코드가 국방메타데이터관리시스템(MDR)의 표준 코드일 경우 'Y'를 기술하고 그렇지 않을 경우 'N'을 기재한다.
- 비교
그 밖의 또 다른 항목이 존재하는 경우 해당 내용을 기재한다.

1.7.4. 세부코드**

- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재한다.
- 코드명
코드를 대표하는 명칭을 기재한다.
- 코드값
코드를 구성하는 세부 코드의 실제 값을 기재한다.
- 코드값명
세부 코드값이 의미하는 명칭을 기재한다.
- 상위 코드명
해당 코드가 계층코드인 경우, 상위분류에 해당되는 코드명을 기재한다.
- 상위 코드값
해당 코드가 계층코드인 경우, 상위분류에 해당되는 코드의 실제 값을 기재한다.
- 상위 코드값명
상위 코드값이 의미하는 명칭을 기재한다.

1.8. 타 시스템간 인터페이스

1.8.1. 인터페이스 구성도

시스템간의 인터페이스를 구체화한 개념도이며, 필요시 체계의 주요 기능을 식별하여 작성한다.

1.8.2. 인터페이스*

- 인터페이스명
시스템간 인터페이스를 구분할 수 있는 명칭을 기재한다.
- 설명
인터페이스에 대한 설명을 기술한다.
- 출발 시스템명
인터페이스 데이터를 송신하는 시스템 명칭을 기재한다.

현행 시스템 분석 작업(1R12)

- 도착 시스템명
인터페이스 데이터를 수신하는 시스템 명칭을 기재한다.

1.8.3. 인터페이스 데이터*

- 인터페이스명
시스템간 인터페이스를 구분할 수 있는 명칭을 기재한다.
- 인터페이스 데이터명
해당 인터페이스에서 교환되는 데이터의 명칭을 기재한다.
- 설명
교환되는 데이터에 대한 설명을 기술한다.
- 상호운용성 달성가능수준
시스템간 데이터 교환에 있어 달성되거나 달성가능한 상호운용성 수준을 기재한다. 세부 구분은 '1.1. 현행 시스템 개요'의 항목설명을 참조한다.
- 주기
시스템 데이터 교환의 처리 주기를 기재한다.

구분	1초 / 1분 / 1시간 / 1일 / 1주일 / 1달 / 1년 / 이벤트발생 시 / 수시로 / 기타(직접기재)
----	---

- 처리량
초당 데이터 처리량을 기재한다. (예 : 10kb/s, 1mb/s 등)
- 적시성
시스템 또는 시스템 기능을 만족하기 위한 필요 시간의 정도를 기재한다.

구분	1일 / 1시간 / 1달 / 8시간 / 보통 / 천천히 / 근실시간 / 실시간 / 기타(직접기재)
----	--

- 보안등급
'보안업무훈령'에 정의된 보안등급을 기재한다.

구분	평문 / 대외비 / III급 / II급 / I급 / SI(특수정보) / 기타(직접 기재)
----	---

- 통신매체 유형
데이터를 교환하기 위한 통신 매체의 유형을 기재한다.

구분	전화선 / 자기매체(디스크 등) / 라디오 / 인공위성 / 데이터베이스 / 광케이블 / UTP / 동축케이블 / HFC / 수동 / 기타(직접 기재)
----	---

- 크기
데이터의 크기를 기재한다.

■ 현행 시스템 분석 작업(1R12)

- 단위
데이터의 측정 단위를 기재한다.

구분	BITS / BYTES / KB / MB / GB / TB / 기타(직접 기재)
----	--

- 포맷유형
데이터/정보의 레이아웃이나 순서와 정렬을 의미하며, 텍스트, 동영상, 오디오, 문서 등을 기재한다.

구분	오디오 / 텍스트 / 팩시밀리 / 이미지 / 동영상 / 규칙 없는 데이터 / ASCII / 비트 데이터 / 면담 / 문서 / 필름 / 자기테이프 / 광디스크 / 자기디스크 / 비디오 / 기타(직접 기재)
----	---

- 트리거 이벤트
시스템 데이터 교환을 트리거하는 이벤트에 대한 간단한 설명을 기술한다.

2. 성능요소

시스템 내 하드웨어, 시스템 소프트웨어, 통신장비 및 기능 등의 특성에 따른 성능요소(예 : 가용성, MTBF(Mean Time Between Failure), 회선 이용률 등) 혹은 성능항목을 선정하고 측정을 통해 시스템의 현재 성능에 대하여 기술한다. 현행 시스템이 성능상의 문제점이 존재하거나 개선이 필요한 경우가 아니라면 생략 가능하다.

3. 문제점 및 해결방안

- 문제점명
현행 시스템의 문제점을 식별할 수 있는 명칭을 기재한다.
- 설명
현행 시스템의 문제점을 기술한다.
- 해결방안
해당 문제점에 대한 해결방안을 기술한다.

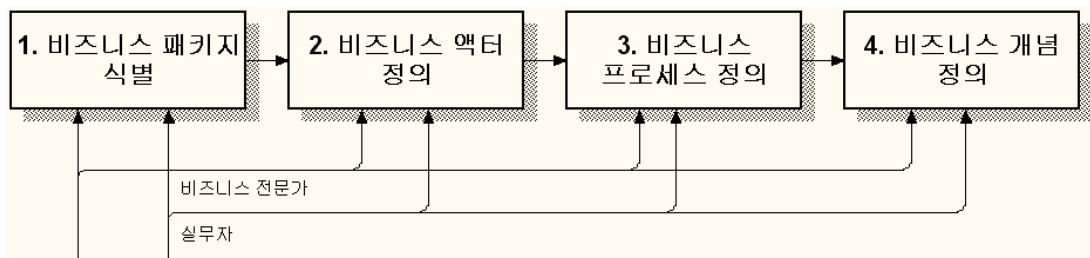
비즈니스 모델링 작업(1R13)

개요

시스템이 지원해야 할 표준 업무를 명확히 하고 요구사항의 이해도를 높이기 위해 개발 대상 시스템이 지원해야 하는 업무 흐름과 주요 역할자 및 정보를 기존 자료 및 사용자, 실무자와의 의사소통을 통해 정의하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 훈령 및 지침 • 국방정보기술아키텍처 • 개념연구 문서 • 현행 관련 문서
산출물	<ul style="list-style-type: none"> • 비즈니스 정의서
참여자	<ul style="list-style-type: none"> • 비즈니스 전문가 개발 대상 시스템이 지원해야 할 업무를 이해하고 모델링한다. • 사용자 개발 대상 시스템이 지원해야 할 업무를 수행하는 담당자로 업무의 이해를 돕고 작성 결과를 확인한다.
관련기법	<ul style="list-style-type: none"> • UML • 브라운 페이퍼 분석
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 비즈니스 패키지 식별

개발 대상 시스템이 지원해야 할 업무를 비즈니스 패키지로 식별하고 업무 절차인 비즈니스 프로세스를 식별한다. 비즈니스 패키지의 식별 및 비즈니스 프로세스 식별의 완전성을 높이기 위해 비즈니스 액터 정의를 병행해 수행할 수 있다.

비즈니스 모델링 작업(1R13)

- 비즈니스 패키지 식별
비즈니스 프로세스를 정의하기 전에 어떤 단위로 업무를 분류할 것인지를 결정해 비즈니스 패키지를 식별한다.
- 비즈니스 프로세스 식별
최하위 비즈니스 패키지별로 해당 비즈니스 패키지에서 수행하는 업무 절차를 비즈니스 프로세스로 식별한다. 비즈니스 프로세스는 특정 임무 또는 목표를 달성하기 위해 해당 비즈니스 프로세스를 주관하는 조직의 담당자가 다른 조직 또는 담당자, 외부 시스템 등과 상호작용하여 결과물을 생성하는 일련의 활동들의 묶음이다.

2. 비즈니스 액터 정의

업무를 수행하는 비즈니스 액터를 정의한다. 비즈니스 액터는 비즈니스 패키지 식별과 병행해 정의하며, 비즈니스 프로세스 정의시 이를 정제 한다.

- 비즈니스 액터 정의
업무를 수행하는 내/외부 조직 또는 업무 수행자, 업무와 관련 있는 외부 시스템 등 비즈니스 프로세스에 참여하는 모든 역할을 실제 조직 또는 담당자 관점에서 묶은 비즈니스 액터를 정의한다. 비즈니스 액터는 업무와 관련된 정보를 생성, 참조 또는 소비해 업무를 처리하는 업무 참여자이다.

3. 비즈니스 프로세스 정의

임무 또는 업무 목표를 달성하기 위해 요구되는 업무 흐름을 식별해 비즈니스 프로세스로 정의한다. 비즈니스 프로세스는 업무를 모델링 하므로 개발 대상의 경계를 설정하는 기초 자료로 활용할 수 있다.

- 비즈니스 프로세스 정의
비즈니스 패키지 식별에서 식별된 비즈니스 프로세스를 상세화/구체화하고 정제한다.
- 비즈니스 액션 정의
식별된 비즈니스 액터가 비즈니스 프로세스에서 수행하는 작업을 비즈니스 액션으로 식별하고 비즈니스 액션간 관계를 찾아 비즈니스 액션간 정보의 흐름인 비즈니스 액션 흐름으로 식별한다.
- 비즈니스 프로세스 및 액터 정제
비즈니스 프로세스 정의를 수행하면서 비즈니스 액터를 지속적으로 정제하고 비즈니스 프로세스간 연속성이 적고 관련 업무를 구체적으로 파악할 수 있는 수준으로 작성한다.

■ 비즈니스 모델링 작업(1R13)

비즈니스 프로세스가 시작되기 위한 사전조건과 주요 입력물(비즈니스 개념), 그리고 종료된 후 달성된 업무의 모습, 생성된 정보 등 사후 조건과 주요 결과물(비즈니스 개념)을 식별하고 하나의 비즈니스 액션에서 조건에 따라 서로 다른 행위를 수행할 경우 이를 비즈니스 규칙으로 정의한다.

4. 비즈니스 개념 정의

비즈니스 프로세스에서 생성, 참조, 소비되는 정보를 비즈니스 개념으로 정의한다. 비즈니스 개념은 비즈니스 액션을 통해 생성된 단일 정보나 정보의 묶음이 될 수 있으며, 비즈니스 프로세스 정의에서 식별한 주요 입력물 및 결과물이 될 수 있다.

- 비즈니스 개념 정의

비즈니스 액션간에 전달되는 정보를 식별하고 각 정보 중 업무 수행에 있어 중요한 정보를 모아 비즈니스 개념을 정의한다. 비즈니스 개념에 포함되어야 할 주요 항목과 각 항목의 특징 및 제약 조건을 식별해 비즈니스 개념을 정의한다.

■ 작업 수행시 고려사항

■ 비즈니스 모델링 작업의 수행 여부

- 비즈니스 모델링 작업은 개발 대상 시스템이 지원해야 할 업무 범위를 명확히 하고 업무 수행에 따른 주요 행위자, 업무 처리 흐름, 입출력 정보를 정의하는데 목적이 있으므로 수집한 자료가 위의 사항을 만족하면 기존 자료를 참고하는 것으로 비즈니스 모델링 작업을 대체할 수 있다.

■ 비즈니스 패키지 식별시 고려사항

- 비즈니스 패키지는 업무 분류이므로 기존에 식별된 비즈니스 패키지가 있다면 참고한다. 만일 없을 경우 업무를 분할해 비즈니스 패키지를 식별하되, 업무적 연관성이 적고 업무와 관련된 자료/정보의 응집도가 높도록 식별한다.

■ 비즈니스 액터 정의시 고려사항

- 비즈니스 액터는 가능한 직관적인 이해가 가능하도록 정의하고 실제 조직, 담당자 또는 시스템과의 관계가 식별되어야 한다. 동일한 업무를 다수의 조직 또는 담당자가 수행할 경우 논리적으로 통합한 비즈니스 액터를 정의할 수 있다.

비즈니스 모델링 작업(1R13)

■ 비즈니스 프로세스 정의시 고려사항

- 비즈니스 프로세스는 시스템화 대상인 업무를 식별해 정의한 것으로 사용자 관리, 타 시스템과의 연동 등 시스템 운용에 필요한 흐름은 정의하지 않는다. 또한 비즈니스 프로세스는 업무 관점에서 작성하는 것으로 시스템의 기능 또는 시스템 처리 절차에 따라 정의하지 않는다.
- 비즈니스 액션은 한 비즈니스 액터가 입력물을 받아 처리과정을 거쳐 결과물을 생성하는 단위로 비즈니스 액션의 작업량, 수행 시간 등 처리량 보다는 주요한 정보를 생성, 소비하는 관점에서 식별한다.
- 비즈니스 프로세스는 다양한 관점의 요구사항을 이해하고 개발할 시스템의 범위를 명확히 하기 위한 것으로 업무의 세부적인 표현 보다는 범위를 명확히 할 수 있는 수준으로 작성한다. 시스템상에서 처리되어야 할 업무의 상세 흐름은 이후 유스케이스 모델링 작업(1R31)에서 작성한다.

■ 비즈니스 개념 정의시 고려사항

- 비즈니스 개념은 업무와 관계된 주요 정보를 중심으로 정의하며, 업무 자체에 대한 개념이나 시스템 운용과 관련된 정보는 정의하지 않는다.
- 비즈니스 개념 정의시 주요 비즈니스 개념과 비즈니스 개념간의 관계 등 해당 비즈니스 프로세스와 비즈니스 개념을 이해하기 용이한 수준으로 작성한다. 또한, 비즈니스 개념은 업무의 입/출력물을 식별하기 위한 것으로 시스템 관점에서의 정규화 또는 일반화를 수행하지 않는다.

■ 비즈니스 모델링시 표기법

- 비즈니스 모델링을 위한 표기법은 매우 다양하며, 사업의 특성, 표기법이 표현하는 정보를 고려해 선택한다. 예로 비즈니스 패키지 식별시 UML의 패키지 다이어그램, 구조도 등을 이용할 수 있다.
- 비즈니스 프로세스 정의시 UML의 액티비티 다이어그램이나 IDEF0를 이용할 수 있으며, 업무의 범위를 명확히 하는데는 UML의 유스케이스 다이어그램의 표기법을 이용할 수 있다.
- 비즈니스 개념은 UML의 클래스 다이어그램이나 개체관계도를 이용할 수 있다.

■ 비즈니스 모델링 결과와 이후 작업과의 관계

- 비즈니스 패키지는 시스템 아키텍처 정의 작업(1R21)에서 서브시스템을 정의하는데 참고할 수 있으며, 요구사항 분석 활동(1R3)에서 사용자 인터페이스의 메뉴 구성 및 시스템의 기능 분할로 유도된다.

■ 비즈니스 모델링 작업(1R13)

- 비즈니스 액터는 유스케이스 모델링 작업(1R31)에서 시스템과 상호작용하는 액터로 유도된다.
- 비즈니스 프로세스는 요구사항 분석 활동(1R3)에서 유스케이스의 기본 흐름, 사용자 인터페이스 흐름, 객체간 메시지 전달 흐름으로 유도된다. 또한, 유스케이스 식별의 기준으로 활용되며 비즈니스 프로세스와 비즈니스 액터간의 관계는 유스케이스 모델링 작업(1R31)에서 유스케이스와 액터간의 관계로 유도된다.
- 비즈니스 개념은 시스템에서 제공되어야 할 데이터로 유도되어 요구사항 분석 활동(1R3)에서 유스케이스에서 처리되어야 할 데이터, 시스템간 연동되어야 할 데이터, 사용자 인터페이스에 표현되어야 할 항목, 엔티티 클래스 등으로 유도된다.

■ 비즈니스 모델링 구성요소간 관계

- 비즈니스 패키지는 하나 이상의 비즈니스 프로세스를 포함하며, 비즈니스 프로세스는 하나의 비즈니스 패키지에만 포함된다.
- 비즈니스 프로세스는 하나 이상의 비즈니스 액션을 포함하며, 비즈니스 액션은 하나의 비즈니스 프로세스에만 포함된다.
- 비즈니스 액터는 여러 비즈니스 프로세스에서 액션을 수행하며, 비즈니스 프로세스는 여러 비즈니스 액터의 행위로 구성된다. 비즈니스 액터는 여러 비즈니스 액션을 수행하며, 비즈니스 액션은 하나의 비즈니스 액터가 수행한다.
- 비즈니스 액션은 비즈니스 규칙을 포함할 수 있으며, 비즈니스 규칙은 하나의 비즈니스 액션에 포함된다.
- 비즈니스 개념은 여러 비즈니스 프로세스에서 생성, 참조 또는 소비되며, 비즈니스 프로세스는 여러 비즈니스 개념을 생성, 참조 또는 소비한다.
- 비즈니스 액션간 흐름과 비즈니스 개념간 관계는 동등한 수준에서 방향성을 가지며, 비즈니스 패키지간 관계는 상위 비즈니스 패키지가 하위 비즈니스 패키지를 포함하는 관계이다.

비즈니스 모델링 작업(1R13)

[서식] 비즈니스 정의서(1R13a)

1. 비즈니스 패키지

1.1. 비즈니스 패키지 구성도

1.2. 비즈니스 패키지 목록

비즈니스 패키지		비즈니스 패키지 설명	하위 비즈니스 프로세스명
패키지명	수준		

2. 비즈니스 액터

2.1. 비즈니스 액터 목록

비즈니스 액터명	비즈니스 액터 설명	관련 원천

3. (비즈니스 패키지명)

3.1. 비즈니스 프로세스 목록

비즈니스 프로세스명	비즈니스 프로세스 설명

3.2. (비즈니스 프로세스명)

3.2.1. 비즈니스 프로세스도

3.2.2. 비즈니스 규칙

비즈니스 액션명	비즈니스 규칙명	비즈니스 규칙 설명

4. 비즈니스 개념

4.1. 비즈니스 개념 구성도

비즈니스 모델링 작업(1R13)

4.2. 비즈니스 개념 목록

비즈니스 개념명	비즈니스 개념 설명	관련 비즈니스 프로세스	
		프로세스명	관계 구분

■ 비즈니스 모델링 작업(1R13)

■ [항목 설명] 비즈니스 정의서(1R13a)

1. 비즈니스 패키지

비즈니스 패키지 구성도를 작성하고 비즈니스 패키지 목록을 작성한다.

1.1. 비즈니스 패키지 구성도

비즈니스 프로세스간 관계와 비즈니스 개념을 고려해 비즈니스 패키지 구성도를 작성한다.

1.2. 비즈니스 패키지 목록

- 비즈니스 패키지명
비즈니스 패키지의 명칭을 기재한다.
- 비즈니스 패키지 수준
비즈니스 패키지간의 상하 관계를 나타내며, 최상위 수준을 “1”로 표시하고 하위 수준은 “점_숫자”의 형태로 기재한다.(예 : 1.1, 1.1.21, 1.1.21.1 등)
- 비즈니스 패키지 설명
비즈니스 패키지에서 수행하는 주요 업무, 목표, 관련 조직 등 설명을 기술한다.
- 하위 비즈니스 프로세스명
비즈니스 패키지에 속한 하위 비즈니스 프로세스의 명칭을 기재한다.

2. 비즈니스 액터

비즈니스 프로세스에 참여하는 비즈니스 액터 목록을 작성한다.

2.1. 비즈니스 액터 목록

- 비즈니스 액터명
비즈니스 액터의 명칭을 기재한다.
- 비즈니스 액터 설명
비즈니스 액터의 주요 임무에 대한 설명을 기술한다.
- 관련 원천
비즈니스 액터로 묶인 내/외부 조직, 담당자 또는 시스템 등의 명칭을 기재한다. 단, 비즈니스 액터가 단일 조직 또는 시스템 등으로 구성되어 비즈니스 액터가 관련 원천이 일치할 경우 작성하지 않는다.

3. (비즈니스 패키지명)

비즈니스 패키지별로 비즈니스 프로세스 목록을 작성하고 각 비즈니스 프로세스별로 흐름도를 작성한다.

■ 비즈니스 모델링 작업(1R13)

3.1. 비즈니스 프로세스 목록

- 비즈니스 프로세스명
비즈니스 프로세스 명칭을 기재한다.
- 비즈니스 프로세스 설명
비즈니스 프로세스의 임무, 업무 목표 및 주요 업무와 비즈니스 개념을 기술한다.

3.2. (비즈니스 프로세스명)

3.2.1. 비즈니스 프로세스도

비즈니스 프로세스별로 비즈니스 액션간의 흐름을 모델링한다.

3.2.2. 비즈니스 규칙

비즈니스 프로세스에서 사용되는 비즈니스 규칙이 있을 경우 해당 비즈니스 규칙을 기술한다.

- 비즈니스 액션명
해당 비즈니스 규칙이 사용되는 비즈니스 액션의 명칭을 기재한다.
- 비즈니스 규칙명
비즈니스 규칙의 명칭을 기재한다.
- 비즈니스 규칙 설명
비즈니스 규칙을 의사코드(pseudo-code), 정형명세(formal method), 자연어 등으로 기술한다.

4. 비즈니스 개념

비즈니스 개념 및 비즈니스 개념간 관계를 비즈니스 구성도로 작성하고 비즈니스 개념 목록을 작성한다.

4.1. 비즈니스 개념 구성도

비즈니스 개념 및 비즈니스 개념간 관계를 비즈니스 개념 구성도로 작성한다.

4.2. 비즈니스 개념 목록

- 비즈니스 개념명
비즈니스 개념의 명칭을 기재한다.
- 비즈니스 개념 설명
해당 비즈니스 개념의 설명과 비즈니스 개념을 구성하는 주요 항목을 기술한다.

비즈니스 모델링 작업(1R13)

- 관련 비즈니스 프로세스명
해당 비즈니스 개념을 생성, 참조 및 소비하는 비즈니스 프로세스명을 기재한다.
- 관련 비즈니스 프로세스 관계 구분
해당 비즈니스 개념과 비즈니스 프로세스간의 관계를 생성, 참조, 수정 등으로 기술한다.

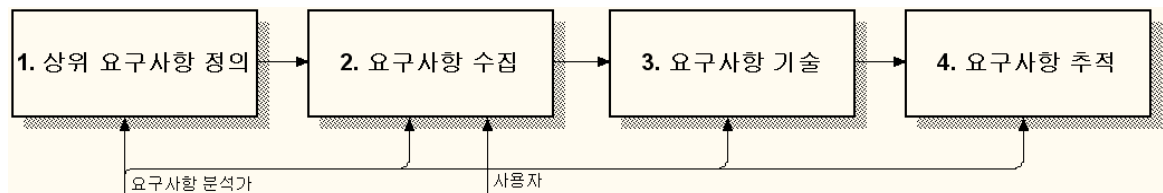
요구사항 정의 작업(1R14)

개요

개발 대상 시스템이 이루고자 하는 목표와 상위수준의 요구사항을 정의한 후, 사용자와의 면담, 설문 및 워크숍 등을 수행하여 상세한 요구사항을 도출하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 계약 관련 문서 훈령 및 지침 국방정보기술아키텍처 개념연구 문서 현행 시스템 분석서 비즈니스 정의서
산출물	<ul style="list-style-type: none"> 요구사항 정의서, (부록)요구사항 추적표
참여자	<ul style="list-style-type: none"> 요구사항 분석가 사용자의 요구사항을 수집해 개발 대상 시스템의 요구사항을 정의한다. 사용자 개발 대상 시스템에 대한 요구사항을 제기한다.
관련기법	<ul style="list-style-type: none"> 마인드 맵 면담 브라운 페이퍼 분석 브레인 라이팅 브레인 스토밍 설문 파레토 다이어그램 피시본 다이어그램

절차 흐름도



■ 요구사항 정의 작업(1R14)

■ 절차 설명

1. 상위 요구사항 정의

프로젝트 수행의 기준이 되는 프로젝트 목표를 정의하고 제안요청서, 개념연구 문서 등 수집한 자료를 분석해 상위 수준의 요구사항을 정의한다.

- 프로젝트 목표 정의
개발 대상 시스템의 목적을 기반으로 프로젝트의 목표를 정의한다. 프로젝트의 목표는 일차적으로 개념연구 문서 또는 제안요청서 등을 통해 제시된다. 본 작업에서는 현행시스템 분석, 비즈니스 정의 등을 통해 변경 또는 보완된 최종적인 프로젝트의 목표에 대해 정의한다.
- 상위 요구사항 정의
프로젝트의 목표를 달성하거나 업무를 수행함에 있어 정보시스템이 갖추어야 할 필요한 조건이나 능력을 만족시키기 위한 상위 수준의 기능적 요구사항과 비기능적 요구사항을 기술한다.

2. 요구사항 수집

사용자 및 선행 결과물을 기반으로 개발 대상 시스템의 요구사항을 수집하고 필요한 자료를 획득한다.

- 요구사항 수집 준비
개발 대상 시스템과 관련된 조직 및 사용자를 파악하고 면담 또는 워크숍 등의 계획을 수립한다. 요구사항 수집을 위한 자료 확보 계획을 우선순위에 따라 수립한다.
- 요구사항 수집
계획된 면담과 설문 작업 등을 통하여 사용자의 요구사항을 이해하며, 수집된 정보를 정리/분석 한다.

3. 요구사항 기술

수집된 요구사항을 구체화하여 요구사항을 정의한다.

- 요구사항 분류
수집된 요구사항을 기반으로 개발 대상 시스템에 대한 사용자의 기능적 요구사항 및 비기능적 요구사항을 파악한다.

■ 요구사항 정의 작업(1R14)

- 요구사항 정의
요구사항별로 개발 대상 시스템에 대한 구체적인 요구사항을 정의하여 기술하고 중요도에 따라 우선순위를 할당한다.

4. 요구사항 추적

요구사항은 사용자가 시스템에서 제공되기를 원하는 기능이므로 분석, 설계, 구현 및 테스트 과정을 통해 개발되며, 프로젝트 진행 및 결과의 성공여부를 판단하기 위한 기준이 된다. 따라서 요구사항의 구현 여부를 확인하고 변경 통제를 위한 요구사항 추적을 실시한다.

- 요구사항 추적 관계 작성
요구사항들이 어떤 유스케이스로 명세화되며, 어떤 컴포넌트로 실체화되고 테스트가 수행되는지 확인이 가능하도록 요구사항과 개발 결과간의 추적 관계를 정의한다.
- 요구사항 추적표 작성
정의된 요구사항을 요구사항 추적표에 등재하고 식별된 추적 관계에 따라 요구사항이 구현됨을 관리한다.

■ 작업 수행시 고려사항

■ 상위 요구사항 정의시 고려사항

- 이 작업에서 작성되는 상위 요구사항은 다양한 관점의 요구사항을 반영할 수 있는 개념적인 수준으로 정의해야 하며, 세부 내용은 '요구사항'을 통해 기술되도록 한다.
- 상위 요구사항은 프로젝트의 목표를 달성하거나 업무를 수행함에 있어 시스템이 갖추어야 할 필요한 조건이나 능력으로써, 단위 업무 수준이 아닌 서비스 시스템 혹은 대/중 기능 분류 관점의 요구사항을 의미한다.

■ 자료 수집시 고려사항

- 자료 수집시 가급적 실제 문서나 서식을 수집하는 것이 좋다. 이를 바탕으로 데이터의 유형을 파악할 수 있고 화면 및 보고서의 서식을 정의할 수 있다.
- 실무에 쓰이는 서식이 다양하게 존재하는 경우 사용자와의 협의를 통해 서식을 표준화시킬 것인지 여부를 결정해야 한다.

■ 요구사항 정의 작업(1R14)

■ 설문 계획 작성 및 결과 분석시 고려사항

- 개발 대상 시스템의 다양한 이해관계자를 식별하는데 다음과 같은 기준을 참조한다.
 - 실제 시스템을 사용할 사용자는 누구인가?
 - 시스템의 결과물에 영향을 받는 사람은 누구인가?
 - 시스템을 평가할 사람은 누구인가?
 - 시스템을 유지 보수할 사람은 누구인가?
- 사용자의 설문 분석을 실시한 경우, 결과는 차트나 그래프 형태로 도식적으로 표현하여 결과의 의미를 쉽게 이해할 수 있도록 한다.

■ 요구사항 기술시 고려사항

- 요구사항은 상위 요구사항에 준하여 기술하는 것을 원칙으로 하며 시스템 성능, 사용자 인터페이스, 자료구축 및 보안 등과 같은 시스템 개발과 관련된 모든 요구사항도 포함하여 기술한다.
- 일반적으로 사용자가 명확하게 제시하지 못하지만 향후 시스템 운용을 위해 필요한 요구사항에 대해서는 사용자와의 협의를 통해 유도하고, 정의하는 것이 바람직하다.
- 업무 전문가와 개발팀이 협의하여 요구사항의 중요도에 따라 우선순위를 할당한다.
 - 해당 요구사항이 실제 업무에서 얼마나 빈번하게 사용되는가?
 - 해당 요구사항이 즉시 구현되지 않는다면 그 프로젝트의 비즈니스 목표에 중대한 영향이 있는가?
 - 해당 요구사항이 다른 요구사항의 선행 작업으로 연결되어 있는가?
 - 해당 요구사항을 해결할 수 있는 다른 방법이 있는가?

■ 요구사항 추적표 작성

- 요구사항 추적표는 일반적으로 요구사항을 기준으로 유스케이스나 사용자 인터페이스, 컴포넌트 등이 어떻게 매핑되는지를 확인할 수 있도록 작성한다.
- 요구사항 추적표는 정형화된 서식이 존재하는 것은 아니므로 프로젝트 및 시스템의 특성에 따라 적절한 서식을 정의하되, 요구사항을 기준으로 각 항목을 매핑할 것인지, 추적표상의 항목을 기준으로 단계별로 추적이 가능하도록 매핑할 것인지 등을 고려한다.

■ 요구사항 정의 작업(1R14)

- 요구사항 추적표는 시스템 개발 전 과정에서 현행화하여 관리해야하며, 최종적인 결과가 요구사항 정의서의 부록으로 제출되어야 한다.
- 변경 통제에 따라 요구사항이 삭제되었을 경우 요구사항 추적표에서 해당 요구사항을 삭제하지 않고 이력을 관리하도록 한다.
- 요구사항 추적표는 요구사항을 기준으로 시스템 개발 과정에서의 추적을 목적으로 하지만 필요시 인도의 용이성이나 사업관리의 용이성 등을 고려하여 제안요청서, 제안서, 기술 협상 단계부터 추적할 수도 있다.

■ 요구사항 정의 이후 작업과의 관계

- 요구사항 정의 작업에서 정의된 요구사항을 기반으로 이후 모든 작업이 수행된다. 요구사항의 변경, 추가시 변경 통제 절차에 따라 해당되는 부분에 한해 요구사항 정의 작업부터 재수행하는 것이 필요하다.

■ 요구사항 정의 작업(1R14)

■ [서식] 요구사항 정의서(1R14a)

1. 프로젝트 목표

2. 상위 요구사항

식별자	상위 요구사항명	설명	관련근거

3. 요구사항

식별자	요구사항명	유형	설명	전제조건	관련부서	우선 순위	관련 상위 요구사항

요구사항 정의 작업(1R14)

■ [항목 설명] 요구사항 정의서(1R14a)

1. 프로젝트 목표

개발 대상 시스템의 목적을 만족하기 위해 프로젝트에서 달성해야 할 목표를 기술한다.

2. 상위 요구사항

- 식별자
해당 상위 요구사항의 식별자를 기재한다.
- 상위 요구사항명
상위 요구사항 내용을 축약하여 명칭으로 기재한다.
- 설명
상위 요구사항에 대하여 구체적으로 기술한다.
- 관련근거
상위 요구사항을 도출하는데 근거가 된 제안요청서, 개념연구 문서, 과업내역서, 사업수행계획서, 제공자료 및 설문/면담 결과물 등을 식별하여 기재한다.

3. 요구사항

상위 요구사항을 기반으로 사용자와의 면담, 설문 및 워크숍 등을 수행하여 상세한 요구사항을 도출하고 이를 기술한다.

- 식별자
해당 요구사항의 식별자를 기재한다.
- 요구사항명
요구사항 내용을 축약하여 명칭으로 기재한다.
- 유형
업무 기능에 대한 요구사항일 경우에는 '기능'을 기재하고 성능, 안전, 보안 등 시스템 운용을 위한 요구사항이나 조작, 화면 구성 등 사용자 조작을 위한 요구사항일 경우에는 '비기능'을 기재한다.
- 설명
사용자의 요구사항을 구체적이고 상세하게 기술한다.
- 전제조건
사용자 요구사항이 구현되기 위해 선행되어야 하는 사항을 기술한다.
- 관련부서
사용자 요구사항과 관련된 모든 조직을 기재하며 비즈니스 정의서에서 식별된 조직명을 기재한다.

■ 요구사항 정의 작업(1R14)

- 우선순위
요구사항의 중요도에 따라 우선순위를 '상', '중', '하'로 기재한다.
- 관련 상위 요구사항
해당 요구사항과 추적이 가능하도록 관련된 상위 요구사항의 식별자를 기재한다.

■ 요구사항 정의 작업(1R14)

■ [서식] 부록 - 요구사항 추적표

1. 요구사항 추적표

요구사항		유스케이스		사용자 인터페이스		컴포넌트		시스템 통합 테스트케이스	
식별자	요구사항명	식별자	유스케이스명	식별자	화면명	식별자	컴포넌트명	테스트케이스 ID	테스트케이스명

■ 요구사항 정의 작업(1R14)

■ [항목 설명] 부록 - 요구사항 추적표

1. 요구사항 추적표

- 요구사항-식별자
해당 요구사항의 식별자를 기재한다.
- 요구사항명
요구사항 명칭을 기재한다.
- 유스케이스-식별자
요구사항에 매핑되는 유스케이스의 식별자를 기재한다.
- 유스케이스명
유스케이스 명칭을 기재한다.
- 화면-식별자
유스케이스에 매핑되는 사용자 인터페이스, 즉 화면의 식별자를 기재한다.
- 화면명
화면의 명칭을 기재한다.
- 컴포넌트-식별자
유스케이스에 매핑되는 컴포넌트의 식별자를 기재한다.
- 컴포넌트명
컴포넌트의 명칭을 기재한다. 컴포넌트명을 식별자로 사용할 경우 생략하거나 한글명을 기재한다.
- 테스트케이스ID
각 테스트케이스의 ID를 기재한다.
- 테스트케이스명
테스트케이스의 명칭을 기재한다.

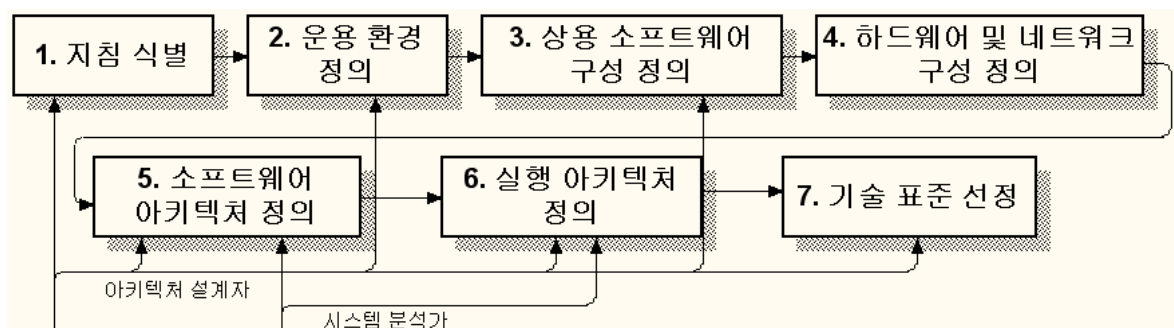
시스템 아키텍처 정의 작업(1R21)

개요

개발 대상 시스템의 요구사항이 충족되도록 시스템 아키텍처를 설계하고 이를 개발자간 공유하기 위해 시스템의 제반 환경, 시스템 아키텍처의 구성 요소 및 동작 방식 등을 정의하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 훈령 및 지침 • 현행 시스템 분석서 • 비즈니스 정의서 • 요구사항 정의서
산출물	<ul style="list-style-type: none"> • 시스템 아키텍처 정의서
참여자	<ul style="list-style-type: none"> • 시스템 분석가 개발 대상 시스템에 대한 성능 및 구현 여부를 고려하여 아키텍처를 평가한다. • 아키텍처 설계자 소프트웨어 아키텍처를 도출하고 최종 시스템의 구현 모습인 시스템 아키텍처를 정의한다.
관련기법	<ul style="list-style-type: none"> • 디자인 패턴 • UML • 아키텍처 비교 분석 기법 • 아키텍처 스타일
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2 • 국방정보기술 표준(DITA)

절차 흐름도



시스템 아키텍처 정의 작업(1R21)

절차 설명

1. 지침 식별

개발 대상 시스템에 영향을 미치는 규정 및 지침을 식별하여 기재한다.

- 지침 식별

현 시점을 기준으로 업무 처리 방법, 시스템의 기능 정의 또는 보안사항 적용 등에 있어 시스템이 준수해야 하는 각 법률, 시행령, 시행규칙, 군 훈령, 예규, 지시, 지침서 등을 식별한다.

2. 운용 환경 정의

개발 대상 시스템이 실제 운용되는 환경을 정의한다.

- 운용 환경 정의

개발 대상 시스템의 운용환경이 통합 환경인지 아니면 분산 환경인지에 초점을 두고 세부 사항을 정의한다. 분산된 환경에서 운용될 경우 개별 환경에 대해 각각 정의하고 이들 간의 관계를 명확히 식별하기 위해 구성도를 작성한다. 만일 분산된 데이터 운용 환경이라면 데이터 무결성 확보를 위한 전략을 수립하여 아키텍처에 반영해야 한다.

3. 상용 소프트웨어 구성 정의

시스템을 구축하기 위해 소요되는 패키지 또는 시스템 소프트웨어를 정의한다. 제안요청서 및 제안서에 기술된 내용을 바탕으로 작성하되, 기 도입된 소프트웨어를 활용할 경우 이에 대한 명시도 함께 한다.

- 패키지 소프트웨어 구성 정의

개발 대상 시스템에서 사용할 패키지 소프트웨어를 식별하고 패키지 소프트웨어의 구성을 정의한다. 패키지 소프트웨어란 커스터마이징을 통해 사용자가 목표로 하는 응용 소프트웨어를 구축할 수 있는 형태로 판매되는 상용 소프트웨어를 말한다.

- 시스템 소프트웨어 구성 정의

개발 대상 시스템의 하드웨어 운영 및 기반 환경과 관련된 시스템 소프트웨어를 식별하고 구성을 정의한다. 시스템 소프트웨어란 운영체제 등과 같이 시스템의 플랫폼을 담당하거나 그래픽 등과 같이 응용 소프트웨어의 일부 기능을 대신하는 상용 소프트웨어를 말한다.

시스템 아키텍처 정의 작업(1R21)

4. 하드웨어 및 네트워크 구성 정의

시스템이 운용되는 시스템 노드를 식별하고 시스템을 구축하기 위해 소요되는 하드웨어와 네트워크 및 통신장비를 명세하며 이들 간의 인터페이스를 정의한다.

- 시스템 노드 식별

개발 대상 시스템이 설치되어 운용될 물리적 시설/설비 또는 논리적 위치, 환경 등을 시스템 노드로 식별한다.

- 하드웨어 구성 정의

제안요청서 및 제안서에 기술된 내용을 바탕으로 하드웨어를 식별하고 구성을 정의한다. 기 도입된 하드웨어를 활용할 경우 이에 대한 명시도 함께 한다. 또한 제안요청서 및 제안서에 기술된 내용과 별도로 하드웨어의 용량과 성능이 요구사항에 비해 적정한지에 대한 분석 및 산정 과정을 수행하고 용량 및 성능이 요구사항을 충족할 수 없을 경우, 대안을 제시한다. 분산환경으로 개발 대상 시스템을 구성할 경우 각 시스템 노드별 하드웨어와 분산환경간의 매핑 관계가 식별되어야 한다.

- 네트워크 구성 정의

시스템 노드별로 노드 내부의 하드웨어, 통신장비간의 랜 구성 및 외부와의 네트워크 구성을 정의한다. 네트워크 구성에 대한 파악을 쉽게 할 수 있도록 구성도를 함께 작성한다.

5. 소프트웨어 아키텍처 정의

응용 소프트웨어에 대한 정적 아키텍처 다이어그램을 작성하고 구성 요소를 정의하며 이들을 배치/연결한 패턴 및 스타일에 대해 정의한다.

개발 대상 시스템을 구성하는 서브시스템 또는 구성 모듈별이 아키텍처가 상이하다면 각각의 아키텍처별로 정의한다. (예 : MIS 시스템과 연동 모듈이 별개의 아키텍처일 경우)

- 소프트웨어 아키텍처 정의

개발 대상 시스템의 소프트웨어 아키텍처를 정의한다. 소프트웨어 아키텍처 정의는 요구사항의 반영 및 품질을 달성할 수 있는 아키텍처를 다수 정의하여 평가를 통해 최선의 아키텍처를 선정하는 것을 목적으로 한다.

- 소프트웨어 아키텍처 평가 및 선정

정의된 소프트웨어 아키텍처들을 동일한 기준으로 평가하여 요구사항 및 품질 목표에 가장 적합한 최종 소프트웨어 아키텍처를 선정한다. 만일 패키지 소프트웨어를 적용하였을 경우 해당 아키텍처의 적정성을 검증해야 한다.

시스템 아키텍처 정의 작업(1R21)

6. 실행 아키텍처 정의

시스템을 구성하는 프로세스를 기술 측면(플랫폼)에 유형별로 분류하고 개념적인 수준에서 시스템 실행시의 동적인 흐름을 나타낸다.

개발 대상 시스템을 구성하는 서브시스템 또는 구성 모듈별이 아키텍처가 상이하다면 각각의 아키텍처별로 정의한다. (예 : MIS 시스템과 연동 모듈이 별개의 아키텍처일 경우)

- 실행 아키텍처 정의

프로세스가 런타임 플랫폼에 의해 제공되는 요소와 어떻게 매핑되는지, 그들이 하드웨어 아키텍처에 어떻게 매핑되는지를 나타내어 실행 아키텍처 다이어그램을 작성한다. 실행 아키텍처는 런타임 플랫폼의 관점에서 제어의 흐름과 자원의 효율적인 사용, 병행 실행 및 비동기 이벤트의 처리 등을 나타내고, 또한 병행 실행되는 쓰레드 간의 통신과 동기화를 나타낸다. 이 아키텍처는 시퀀스 다이어그램, 커뮤니케이션 다이어그램, 액티비티 다이어그램과 같은 동적인 다이어그램으로 작성한다.

7. 기술 표준 선정

개발 대상 시스템의 아키텍처가 어떠한 기술로써 구현되는지를 식별한다. 국방 표준 기술을 우선으로 적용하며, 국방 표준이 아닌 적용 기술은 '국방상호운용성관리훈령' 및 'DITA 가이드라인'에 따라 국방 표준으로 등록 요청한다.

- 국방 표준 선정

최신 DITA(Defence Information Technical stAndard)를 기준으로 정의된 아키텍처의 기술 표준을 식별한다.

- 기타 표준 선정

시스템을 구현하기 위해 DITA이외에 별도 국제, 국가 또는 산업기술 표준 등이 필요한 경우에 이를 식별한다. 사업 특성, 시스템 특성에 맞게 적절한 식별 기준을 수립하여 대상을 선별한다.

시스템 아키텍처 정의 작업(1R21)

■ 작업 수행시 고려사항

■ 소프트웨어 아키텍처 정의시 고려사항

- 소프트웨어 아키텍처는 다양한 관점에서 정의되어야 할 경우가 있다. 따라서 본 방법론의 양식을 시스템 성격에 맞게 보완해서 산출물을 작성해야 한다. 다음은 참조할 수 있는 소프트웨어 아키텍처 관점의 예이다.

소프트웨어 아키텍처 설계 프로세스 지침의 소프트웨어 아키텍처 관점 TTAS.KO-11.0047

- 유스케이스 아키텍처 관점(Usecase Architecture view)
사용자 관점에서 인식하는 시스템의 기능을 설명함.
- 논리 아키텍처 관점(Logical Architecture view)
시스템이 사용되어지는 관점에서의 작업을 수행하는 논리적 컴포넌트와 관계를 정의함.
- 구현 아키텍처 관점(Implementation Architecture view)
컴포넌트 아키텍처 모델을 만족시키는 플랫폼 종속적인 구현 관점의 구현 컴포넌트 아키텍처 모델을 정의함.
- 프로세스 컴포넌트 아키텍처관점 (Process Component Architecture view)
소프트웨어 시스템을 구성하는 플랫폼 독립적인 시스템 컴포넌트를 정의하는 것으로, 소프트웨어 개발자 관점에서 시스템을 어떻게 분할할 것인지 나타냄.
- 배치 아키텍처 관점(Deployment Architecture view)
시스템 개발자 관점에서 물리적 시스템의 노드에 해당하는 실행 시스템에 물리적 컴포넌트가 어떻게 대응하는지를 나타냄.

시스템 아키텍처 정의 작업(1R21)

- 개발 시스템의 필요에 따라 분산 구조나 보안 구조, 사용자 인터페이스 구조 등 여러 관점의 구조를 나타내는 아키텍처 다이어그램을 추가로 작성한다.
- 소프트웨어 아키텍처의 정의, 평가 및 최종 아키텍처 선정에 해당되는 일련의 과정은 요구사항을 만족하고 품질 목표를 달성하는 응용 소프트웨어를 개발하는데 중요한 과정으로 이 과정에 대한 산출물은 별도로 작성해 유지하는 것이 좋다. 시스템 아키텍처 정의서에는 선정된 최종 아키텍처만을 기술한다.
- 상위 요구사항 및 요구사항에서 식별된 비기능적 요구사항, 기능적 요구사항을 위한 비기능적 요소 등이 아키텍처 도출 및 평가 기준이 될 수 있으며 ISO/IEC 9126과 ISO/IEC 14598 등 국제 표준 및 기타 아키텍처 품질평가 방법들을 활용하여 평가를 수행하고 최종 아키텍처를 정의한다.
- 아키텍처 정의시 고려해야 할 요소는 다음과 같다.
 - 아키텍처 요구사항을 모두 반영하였는지
 - 향후 시스템의 운영 및 유지보수시 용이성과 확장성이 반영되었는지
 - 다른 시스템을 통해 품질이 검증되었는지
 - 위험 요소와 한계는 무엇인지
- 소프트웨어 아키텍처는 주 응용 소프트웨어의 동작과 같이 제한적인 모습뿐만 아니라 통제 모듈 등 아키텍처 정의가 필요한 모든 서브시스템에 대해서도 누락 없이 정의하여야 하며, 연동 모듈에 대한 정의는 연동 소요 분석 작업(1R32)에서 실시할 수 있다.

■ 아키텍처 사전 정의시의 산출물 작성

- 개발 시스템이 패키지 소프트웨어나 상용 프레임워크의 도입, 기존 시스템의 커스터마이징 등에 따라 아키텍처 정의 문서가 별도로 존재할 경우에는 해당 문서를 필요시 별도로 첨부하고, 본 방법론의 시스템 아키텍처 정의서는 공통성을 위해 서식을 최대한 준수하여 작성한다.

■ 하드웨어 성능/용량 적정성 판단

- 제안 요청서의 제안 요청 내용에 기재된 하드웨어 성능 및 용량은 개념연구 단계에서 개략적인 요구사항을 바탕으로 산정된 것이다. 보다 구체적이며 세부적인 요구사항이 요구사항 식별 활동을 통해 식별되었으면 이를 바탕으로 하드웨어가 개발 대상 시스템의 요구사항과 체계규격을 만족할 수 있을지에 대한 적정성 판단을 해야 한다.
- 다양한 산정법 존재하므로 별도의 방법에 따라 적정성을 판단한다. (예를 들어 한국정보통신기술협회의 ‘정보시스템 하드웨어 규모 산정 지침’ 참조)

■ 시스템 아키텍처 정의 작업(1R21)

■ 프로토타이핑 기법 활용

- 필요시 사용자와 요구 내용의 이해 및 검증, 아키텍처 성능 평가 등의 목적으로 프로토타입을 제작하여 활용할 수 있다.

■ 시스템 아키텍처 정의 결과와 이후 작업과의 관계

- 시스템 아키텍처 정의의 모든 구성 항목은 향후 분석, 설계, 구현, 테스트 작업에 기준으로 적용된다.
- 소프트웨어 아키텍처, 실행 아키텍처는 컴포넌트 식별 작업(2D11)에서 컴포넌트 목록, 컴포넌트 아키텍처 다이어그램으로 구체화되어 적용된다.

시스템 아키텍처 정의 작업(1R21)

[서식] 시스템 아키텍처 정의서(1R21a)

1. 개요

1.1. 관련지침

지침명	설명	분야	제정년월	최근 개정년월	제정기관

1.2. 개발 대상 시스템 정의

시스템명		상위 시스템명	
설명			
약어		관련 시스템 코드	
사용환경		분산구조	
상호운용성 도달가능수준		도메인명	
운용조직		사용조직	

시스템 아키텍처 정의 작업(1R21)

2. 운용 환경

2.1. 개요

운용 환경명	설 명	관련 개발 대상 시스템명

2.2. 운용 환경 구성도

3. 하드웨어 구성

3.1. 하드웨어 구성도

3.2. 하드웨어

하드웨어명	획득방법				
설명					
장비유형		용도		수량	
Port 수		Disk용량		메모리	
CPU종류		CPU수량		CPU속도	
제조사		모델명		OS	
관련 운용 환경명					

시스템 아키텍처 정의 작업(1R21)

4. 네트워크 및 통신장비 구성

4.1. 네트워크 구성도

4.2. 시스템 노드

시스템 노드명	설 명

4.3. 네트워크

네트워크명	설 명	유형	속도	용도	보안등급	관련 시스템 노드

4.4. 랜

랜명	설 명	속도	관련 통신장비	관련 하드웨어

시스템 아키텍처 정의 작업(1R21)

4.5. 통신장비

통신장비명	설 명	유형	수량	모델명	제조사	관련 하드웨어	시스템 노드

5. 소프트웨어 구성

5.1. 패키지 소프트웨어

소프트웨어명	설 명	버전	제조사

5.2. 시스템 소프트웨어

하드웨어명	유형	소프트웨어명	설명	버전	제조사

시스템 아키텍처 정의 작업(1R21)

6. 아키텍처

6.1. 소프트웨어 아키텍처

6.1.1. 소프트웨어 아키텍처 스타일

6.1.1.1. 아키텍처 다이어그램

6.1.1.2. 아키텍처 설명

아키텍처 설명	관련 요구사항ID

6.1.2. 아키텍처 구성 요소

구분	아키텍처 구성 요소명	설 명	관련 운용 환경명

시스템 아키텍처 정의 작업(1R21)

6.2. 실행 아키텍처

6.2.1. 실행 아키텍처 스타일

6.2.2. 아키텍처 구성 요소

구분	아키텍처 구성 요소명	설 명	관련 소프트웨어 아키텍처 구성 요소명

7. 기술 표준

7.1. 국방 표준

대분류	중분류	표준 번호	표준명	상태
사용자 인터페이스	그래픽 서비스			
	사용자 인터페이스 서비스			
데이터 관리 및 교환	국제화 서비스			
	데이터 관리 서비스			
	데이터 교환 서비스			
체계간 연동	체계간 연동 서비스			
플랫폼 및 서비스	물리 환경 서비스			
	소프트웨어 공학 서비스			
	시스템 관리 서비스			
	운영체제 서비스			

시스템 아키텍처 정의 작업(1R21)

대분류	중분류	표준 번호	표준명	상태
	플랫폼 통신 서비스			
정보보호	정보 보호 서비스			
모델링 및 시뮬레이션	M&S 서비스			

※ 본 서식은 DITA 서비스 분류 체계에 따른 것이므로 DITA의 변화에 따라 서식을 변경해야 함.

7.2. 기타 표준

시스템 아키텍처 정의 작업(1R21)

■ [항목 설명] 시스템 아키텍처 정의서(1R21a)

1. 개요

1.1. 관련지침*

- 지침명
지침을 대표하는 명칭을 기재한다.
- 설명
지침에 대한 설명을 기술한다.
- 분야
지침이 속한 대표적인 분야를 기재한다.

구분	시스템관리 / 보안 / 구매 / 운영관리 / 기타(직접기재)
----	-----------------------------------

- 제정년월
지침이 최초에 제정된 년/월을 기재한다.
- 최근 개정년월
지침이 최근에 개정된 년/월을 기재한다.
- 제정기관
지침을 제정한 기관의 명칭을 기재한다.(예 : 국방부, 방위사업청)

1.2. 개발 대상 시스템 정의*

- 시스템명
시스템을 대표하는 명칭을 기재한다.
- 상위 시스템명
시스템을 포함하는 상위 시스템이 있을 경우 이를 대표하는 명칭을 기재한다.
- 설명
시스템에 대한 설명을 기술한다.
- 약어
시스템을 나타내는 약어로 한글약어 또는 영문약어로 표현할 수 있으며, 일반적으로 영문약어로 기재한다.(예 : KJCCS, MIMS 등)
- 관련 시스템 노트
시스템이 운용되는 실질적 노트를 기재한다. (예 : ○○부대 전산실)
- 사용환경
시스템이 불특정 다수에 의해 사용되는지 또는 내부에서 사용되는 시스템인지 등을 식별하기 위한 속성으로 인터넷, 인트라넷, 엑스트라넷 등으로 구분하여 기재한다.

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

시스템 아키텍처 정의 작업(1R21)

구분	인터넷 / 인트라넷 / 엑스트라넷 / 직접기재 / 미정의 / 기타(직접기재)
----	--

- 분산구조

시스템이 운용될 환경의 분산 구조를 개략적으로 기재한다.

구분	호스트기반 / 클라이언트 / 서버기반 / 웹기반 / 홈페이지 / 기타(직접기재)
----	--

- 상호운용성 도달 가능 수준

시스템이 달성되거나 달성가능한 상호운용성 수준(Level of Information System Interoperability)을 기재한다.

구분	0 : Level 0-하위 수준이 없는 격리된 수준 / 1 : Level 1-하위 수준이 없는 연결된 수준 / 1a : Level 1-연결된 수준, 하위수준 a / 1b : Level 1-연결된 수준, 하위수준 b / 1c : Level 1-연결된 수준, 하위수준 c / 1d : Level 1-연결된 수준, 하위수준 d / 2 : Level 2-하위 수준이 없는 기능적인 수준 / 2a : Level 2-기능적인 수준, 하위수준 a / 2b : Level 2-기능적인 수준, 하위수준 b / 2c : Level 2-기능적인 수준, 하위수준 c / 3 : Level 3-하위수준이 없는 도메인 수준 / 3a : Level 3-도메인 수준, 하위수준 a / 3b : Level 3-도메인 수준, 하위수준 b / 3c : Level 3-도메인 수준, 하위수준 c / 4 : Level 4-하위 수준이 없는 엔터프라이즈 수준 / 4a : Level 4-엔터프라이즈 수준, 하위수준 a / 4b : Level 4-엔터프라이즈 수준, 하위수준 b / 4c : Level 4-엔터프라이즈 수준, 하위수준 c / 기타(직접기재)
----	--

- 도메인명

논리적으로 유사한 시스템의 그룹인 시스템 도메인의 명칭을 기재한다.

- 운용조직

개발 대상 시스템을 운용할 조직명을 기재한다.

- 사용조직

개발 대상 시스템을 사용할 조직명을 기재한다.

시스템 아키텍처 정의 작업(1R21)

2. 운용 환경

2.1. 개요

개발 대상 시스템이 운용되는 환경을 기술하되, 분산 환경에서 운용되면 해당 분산 환경을 모두 식별한다.

- 운용 환경명
개발 대상 시스템이 운용되는 환경의 명칭을 기재한다.
- 설명
운용 환경에 대한 구체적인 설명을 기술한다.
- 관련 개발 대상 시스템명
운용 환경에 관련이 있는 개발 대상 시스템명을 기재한다.

2.2. 운용 환경 구성도

운용 환경과 개발 대상 시스템의 관계 및 구성을 개략적으로 이해할 수 있는 개념도로 나타낸다.

3. 하드웨어 구성

3.1. 하드웨어 구성도

시스템내의 하드웨어와 각각의 연결 정보를 세부적으로 표현한 개념도로, 체계의 물리적 하드웨어 등에 대한 구성을 전체적으로 표현한다. 4.1 네트워크 구성도와 통합해 작성할 수 있다.

3.2. 하드웨어*

- 하드웨어명
시스템을 구성하는 하드웨어에 대한 명칭을 기재한다.
- 획득방법
재활용, 신규 도입 등 하드웨어를 획득하는 방법을 기재한다.
- 설명
하드웨어에 대한 구체적인 설명을 기술한다.
- 장비유형
하드웨어 장비 유형 정보에 대한 구분하여 기재한다.

구분	개인용 PC / PC급 서버 / 중형서버 / 대형서버 / 디스플레이 / 백업장치 / 기타(직접기재)
----	---

시스템 아키텍처 정의 작업(1R21)

- 용도
하드웨어에 대한 용도를 구분하여 기재한다.

구분	내부업무처리용 (MIS, 문서처리 등) / 시스템 내 응용시스템 서버 / 백업용 서버 / 파일서버 / 보안용 / 기타(직접 기재)
----	--

- 수량
하드웨어 수량에 대해 기재한다.
- Port수
하드웨어가 보유하고 있는 총 포트 수를 기재한다.
- Disk용량
하드웨어가 보유하고 있는 디스크 총 용량을 기재한다.
- 메모리
총 메모리 용량을 기재한다.
- CPU종류
하드웨어 CPU 종류를 기재한다.
- CPU수량
하드웨어 CPU 총 개수를 기재한다.
- CPU속도
하드웨어 CPU 속도를 기재한다.
- 제조사
하드웨어에 대한 제조사를 기재한다.
- 모델명
하드웨어의 제조 모델명을 기재한다.
- OS
하드웨어에 탑재된 운영체제를 기재한다.
- 관련 운용 환경명
하드웨어가 운용되는 운용 환경명을 기재한다.

4. 네트워크 및 통신장비 구성

4.1. 네트워크 구성도

네트워크 구성을 가시적으로 확인할 수 있도록 네트워크, 통신장비, 랜, 하드웨어로 구성되는 개념도를 작성하되, 시스템 노드별로 작성한다.
네트워크가 여러 개일 경우 필요시 각각 작성할 수 있다.

시스템 아키텍처 정의 작업(1R21)

4.2. 시스템 노드*

- 시스템 노드명
시스템이 운용되는 실질적 노드를 기재한다. (예 : ○○부대 전산실)
- 설명
시스템 노드에 대한 설명을 기술한다.

4.3. 네트워크*

- 네트워크명
시스템이 운용되는 네트워크 명칭을 기재한다.
- 설명
목적, 규모 등 네트워크에 대한 상세한 설명을 기술한다.
- 유형
네트워크를 구분하기 위한 유형을 기재한다.

구분	LAN / WAN / VPN / ATM / BcN / Internet / Intranet / 백업회선 / 재해복구회선 / 기타(직접 기재)
----	---

- 속도
네트워크의 속도를 기재한다.
- 용도
네트워크의 사용 용도를 기재한다.
- 보안등급
“보안업무훈령”에 정의된 보안등급을 기재한다.

구분	평문 / 대외비 / III급 / II급 / I급 / SI(특수정보) / 기타(직접 기재)
----	---

- 관련 시스템 노드
네트워크가 운용되는 실질적 노드명을 기재한다. (예 : 00부대 전산실)

4.4. 랜*

- 랜명
시스템 노드 내에서 통신장비와 하드웨어간 또는 하드웨어간을 연결하는 물리적인 랜의 명칭을 기재한다.
- 설명
랜에 대한 구체적인 설명을 기술한다. (예 : 토폴로지 유형, 규모 등)
- 속도
내부 속도를 기재한다. (예 : 10MBps, 100MBps 등)

시스템 아키텍처 정의 작업(1R21)

- 관련 통신장비
랜에 연결된 통신장비명을 기재한다.
- 관련 하드웨어
랜에 연결된 하드웨어명을 기재한다.

4.5. 통신장비*

- 통신장비명
데이터 혹은 정보의 전송 기능을 가지고 있는 통신장비의 명칭을 기재한다.
예로 네트워크 스위치와 라우터, 위성 등을 포함한다.
- 설명
통신장비의 통신기능에 대한 간단한 설명을 기술한다.
- 유형
통신장비의 유형을 구분하여 기재한다.

구분	방화벽 / 라우터 / 보안장비 / 허브/ 스위치 / 모뎀 / 게이트웨이 / VPN / 기타(직접 기재)
----	---

- 수량
통신장비 수량에 대해 기재한다.
- 모델명
통신장비의 제조 모델명을 기재한다.
- 제조사
통신장비에 대한 제조사를 기재한다.
- 관련 하드웨어
통신장비와 통신하는 하드웨어를 기재한다.
- 시스템 노드
통신장비가 운용되는 실질적 노드명을 기재한다. (예 : ○○부대 전산실)

5. 소프트웨어 구성

5.1. 패키지 소프트웨어*

- 소프트웨어명
ERP, NMS, DW 등 패키지로 판매되는 솔루션 형태의 소프트웨어의 명칭을 기재한다.
- 설명
패키지 소프트웨어에 대한 구체적인 설명을 기술한다.
- 버전
패키지 소프트웨어에 대한 릴리즈를 포함한 버전을 기재한다.

시스템 아키텍처 정의 작업(1R21)

- 제조사
패키지 소프트웨어에 대한 제조사를 기재한다.

5.2. 시스템 소프트웨어*

- 하드웨어명
시스템 소프트웨어가 탑재되는 하드웨어명을 기재한다.
- 유형
시스템 소프트웨어를 구분하기 위한 유형을 기재한다.

구분	운영체제 / 미들웨어 / DBMS / 문서편집기 / 스프레드시트 / 모델링 툴 / 그래픽툴 / 메일 / 백신 / 보안 / 기타(직접기재)
----	---

- 소프트웨어명
시스템 소프트웨어 명칭을 기재한다.
- 설명
시스템 소프트웨어에 대한 구체적인 설명을 기술한다.
- 버전
시스템 소프트웨어에 대한 릴리즈를 포함한 버전을 기재한다.
- 제조사
시스템 소프트웨어에 대한 제조사를 기재한다.

6. 아키텍처

개발 대상 시스템을 구성하는 서브시스템 또는 구성 모듈의 아키텍처가 상이하다면 「6. 아키텍처」와 「7. 기술표준」을 각각의 아키텍처별로 작성한다.

6.1. 소프트웨어 아키텍처

6.1.1. 소프트웨어 아키텍처 스타일

6.1.1.1. 아키텍처 다이어그램

각 구성 시스템별로 소프트웨어 아키텍처 정의 작업을 통해 구현 관점의 소프트웨어 아키텍처를 개략적 다이어그램으로 작성한다. 아키텍처 스타일 또는 패턴을 적용하여 시스템의 계층을 정의하고 소프트웨어 컴포넌트를 적절한 계층에 할당한 아키텍처를 구성한다.

6.1.1.2. 아키텍처 설명

- 아키텍처 설명
소프트웨어 아키텍처 요소들을 구성하고 결합한 아키텍처 패턴 및 스타일에 대한 설명을 기술한다.

시스템 아키텍처 정의 작업(1R21)

- 관련 요구사항
해당 아키텍처 패턴 및 스타일을 유도한 요구사항을 요구사항 정의서의 요구사항 식별자로 기재한다.

6.2. 아키텍처 구성 요소

- 구분
아키텍처 구성 요소를 묶어 구분할 수 있는 구분자를 기재한다.
- 아키텍처 구성 요소명
아키텍처를 구성하는 요소(컴포넌트 등)명을 각 티어, 레이어 계층으로 구분하여 기재한다.
- 설명
구성 요소의 기능, 특성 등의 설명을 기술한다.
- 관련 운용 환경명
구성 요소가 배치될 운용 환경을 파악하기 위해 관련 운용 환경명을 기재한다.

6.3. 실행 아키텍처

6.3.1. 실행 아키텍처 스타일

각 시스템의 주요 또는 핵심 프로세스를 기술 측면(플랫폼)의 유형별로 분류하여 프로세스가 런타임 플랫폼에 의해 제공되는 요소에 어떻게 매핑되는지와 그들이 하드웨어 아키텍처에 어떻게 매핑되는지를 나타내어 실행 관점에서 개략적 아키텍처 다이어그램을 작성한다.

6.3.2. 아키텍처 구성 요소

- 구분
아키텍처 구성요소를 묶어 구분할 수 있는 구분자를 기재한다.
- 아키텍처 구성 요소명
아키텍처를 구성하는 요소(런타임 플랫폼, 하드웨어 등)명을 구분하여 기재한다.
- 설명
구성 요소의 기능, 특성 등의 설명을 기술한다.
- 관련 소프트웨어 아키텍처 구성 요소명
실행 아키텍처 구성요소에 의존적이거나 물리적으로 구현되는 소프트웨어 아키텍처 구성 요소명을 기재한다.

시스템 아키텍처 정의 작업(1R21)

7. 기술 표준

개발 대상 시스템을 구성하는 서브시스템 또는 구성 모듈의 아키텍처가 상이하다면 「6. 아키텍처」와 「7. 기술표준」을 각각의 아키텍처별로 작성한다.

7.1. 국방 표준

시스템을 구현하기 위한 기술 표준을 DITA(Defence Information Technical stAndard)를 기준으로 작성하고 해당 DITA 버전을 명시한다.

- 대분류
DITA의 대분류명을 기재한다.
- 중분류
DITA의 중분류명을 기재한다.
- 표준 번호
기술 표준 번호(DITA 기준)를 기재한다.
- 표준명
기술 표준명(DITA 기준)을 기재한다.
- 상태
기술 상태 분류를 기재한다.

구분	미래 / 필수 / 사양 / 폐지
----	-------------------

7.2. 기타 표준

시스템을 구현하기 위해 DITA이외에 별도 국제, 국가, 산업기술 표준 등이 필요한 경우에 작성한다. 사업 특성, 시스템 특성에 맞게 적절한 식별 기준을 수립하여 대상을 선별한다.

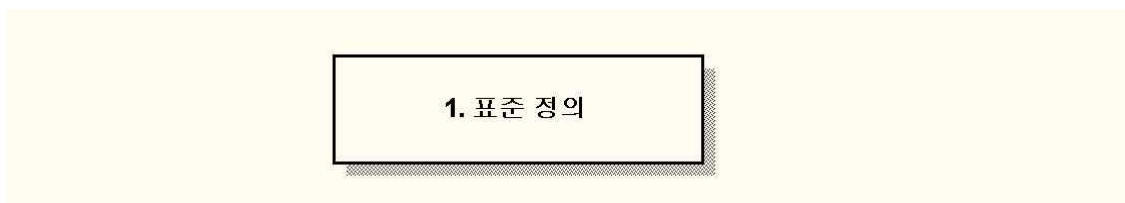
표준 지침 수립 작업(1R22)

개요

시스템 개발 시 준수해야 할 표준 또는 지침을 현재 프로젝트에 맞게 정의하는 작업이다. 사업관리조직 및 개발자들 간의 이해와 공유를 도모하고 산출물의 일관성과 이해성을 높여 시스템 개발이 체계적이고 정확하게 수행되도록 한다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 표준서 및 지침서 상용 S/W 사용자 매뉴얼
산출물	<ul style="list-style-type: none"> 표준 지침서
참여자	<ul style="list-style-type: none"> 표준 작성자 <p>시스템 개발 시 준수해야 할 표준에 대해 정의한다.</p>
관련기법	해당 사항 없음
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 데이터공유환경(SHADE)

절차 흐름도



절차 설명

1. 표준 정의

프로젝트 표준, 사용자 인터페이스 표준, 분석/설계/구현 표준 등 시스템 개발에 적용할 다양한 표준을 정의한다.

- 표준 정의

표준 또는 지침을 현재 프로젝트에 맞게 작성하고 사용자와의 협의를 통해 표준으로 정의하며, 세미나 및 교육 등을 통해 관계자들에게 배포한다.

표준 지침 수립 작업(1R22)

작업 수행시 고려사항

■ 표준 지침 수립 작업의 수행 여부

- 기존의 표준 지침서가 존재하거나 참조 가능한 기존 자료가 있을 경우, 현재 프로젝트에 맞게 수정, 보완작업이 필요한지 검토한 후 사용자와의 협의를 통해 해당 자료를 참고하는 것으로 표준 지침 수립 작업을 대체할 수 있다.

■ 표준 정의시 고려사항

- 시스템 개발은 다양한 이해 관계자가 관여되어 추진된다. 따라서 시스템 개발에 필요한 표준들이 정의되고, 이해 및 공유되어 있지 않으면 원활한 진행이 어려워진다.
- 시스템 개발 과정에서 변경되거나 추가 또는 삭제되는 표준이 존재하는 경우 그 즉시 반영하여 현행화해야 한다.
- 표준을 정의하는데 있어, 국방 분야에 맞게 지정된 표준이 있는지에 대한 검토가 이루어져야 하며, 표준끼리 상충되는 경우가 발생되면 국방 분야의 표준을 우선적으로 적용하는 것을 검토해야 한다.

■ 표준 지침서 작성 방법

- 표준 지침서 서식에서 제시하는 항목 이외에 프로젝트의 특성이나 개발 환경 등 기타 추가적인 표준이 필요하면 추가로 작성하도록 한다. 또한, 서식에 구애받지 않고 각각의 표준에 따라 세부 문서로 나누어 작성하거나 통합하여 작성할 수 있으며 별도의 표준문서가 존재하는 경우, 해당 표준을 준수할 것을 정의하고 참조하도록 기술한다.

표준 지침 수립 작업(1R22)

[서식] 표준 지침서(1R22a)

1. 프로젝트 표준
2. 분석/설계 표준
3. 사용자 인터페이스 표준
4. 데이터 모델링 표준
5. 개발 표준
6. 기타

표준 지침 수립 작업(1R22)

■ [항목 설명] 표준 지침서(1R22a)

1. 프로젝트 표준

프로젝트를 진행하면서 필요한 문서작성 표준, 문서 및 요구사항 등의 관리를 위한 식별자 부여 규칙, 파일 서버 관리 규칙, 백업 규칙 등 프로젝트 전반에 걸쳐 적용되어야 할 표준을 기술한다.

2. 분석/설계 표준

분석 및 설계 단계에 실시하게 되는 유스케이스, 클래스 및 컴포넌트 모델링 등 분석/설계 활동 수행 및 산출물 작성 방법에 대한 표준, 모델링 도구 활용 방법 등에 대한 가이드 또는 지침을 기술한다.

3. 사용자 인터페이스 표준

시스템 전반에 걸쳐 일관된 사용자 인터페이스를 구현하고, 체계적인 개발을 위해 다음과 같은 사용자 인터페이스 표준 항목 등에 대해 기술한다.

- 사용자 인터페이스 디자인 컨셉
- 레이아웃, 네비게이션, 폰트, 칼라 및 그래픽 요소 등에 대한 가이드라인
- 스타일 시트, 버튼, 팝업창 및 에리창 등 공통적인 화면 표준
- 초기 화면, 분기 화면 등 각 화면의 프레임 구조, 메뉴 구조 및 보고서에 대한 표준

4. 데이터 모델링 표준

효과적인 데이터 모델링 및 데이터베이스 구축을 위해 데이터 모델링 작업 절차와 이에 따른 데이터 모델링 도구의 활용 방법 등에 대한 지침을 제시하고 테이블, 인덱스 및 뷰 등 데이터베이스 설계에 관한 표준도 함께 정의한다. 단, 엔티티, 속성 및 테이블 등에 대한 명명규칙은 국방 데이터공유환경(SHADE)을 준수해야 하므로 '국방상호운용성관리 지시'에서 제시된 것과 같이 국방메타데이터관리체계(MDR)에서 관리하고 있는 데이터 표준화 지침서의 명명규칙을 준수해야 한다.

5. 개발 표준

시스템을 구현할 때 필요한 개발 도구 및 서버 S/W 설치, 작업 디렉토리 구조 등과 같은 개발 환경 설정을 위한 지침들과 개발 언어별 표준 코드 또는 구현 예제, 코딩 정책, 명명 규칙 및 주석 지침 등과 같은 구현 표준들을 기술한다.

6. 기타

위에 제시된 표준 이외에 빌드 및 배포 표준, 상용 S/W 사용 가이드 등과 같이 프로젝트에 따라 추가적으로 필요한 표준을 정의하여 기술한다.

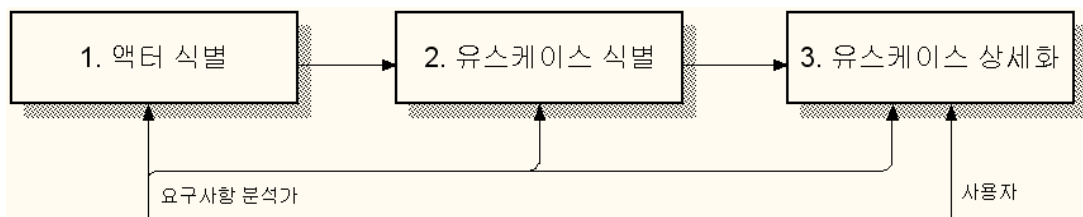
유스케이스 모델링 작업(1R31)

개요

개발자가 개발 대상 시스템의 범위를 명확히 하고 시스템이 제공하는 기능을 분석하기 위해 비즈니스 정의서와 요구사항 정의서를 기반으로 개발 대상 시스템이 제공해야 하는 기능을 액터와 유스케이스로 모델링하고 액터와 유스케이스에 대한 상세한 설명을 기술하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 비즈니스 정의서 요구사항 정의서, (부록)요구사항 추적표
산출물	<ul style="list-style-type: none"> 유스케이스 정의서
참여자	<ul style="list-style-type: none"> 요구사항 분석가 액터와 유스케이스를 모델링한다. 사용자 개발 대상 시스템을 사용해 업무를 수행할 자로 액터와 유스케이스 모델링 결과를 검토한다.
관련기법	<ul style="list-style-type: none"> UML

절차 흐름도



※ 유스케이스 모델링의 각 절차는 한 번씩만 수행하는 것이 아니라 절차 흐름을 계속 반복하면서 수행한다.

절차 설명

1. 액터 식별

시스템과 상호작용하면서 서비스를 받거나 정보를 제공하는 시스템 외부의 논리적 역할자인 액터를 식별한다.

■ 유스케이스 모델링 작업(1R31)

- 개발 대상 시스템 범위 파악
액터를 식별하기 위해 비즈니스 정의서, 요구사항 정의서를 기반으로 개발할 시스템의 범위를 결정한다.
- 액터 식별
시스템과 상호작용하는 개인, 그룹, 조직, 장비 및 연동이 필요한 타 시스템 등을 액터로 식별한다. 식별된 액터에 대해 명칭을 부여하고 액터가 수행하는 역할(시스템과의 상호작용 내역)을 정의한다. 식별된 액터들을 비교해 액터간 혼동이나 역할 중복이 없는지 확인하고 정제한다.

2. 유스케이스 식별

식별된 액터별로 개발 대상 시스템이 제공하거나 제공받아야 할 서비스를 유스케이스로 식별한다.

- 유스케이스 식별
개발 대상 시스템이 액터에게 제공해야 할 서비스와 제공 받아야 할 서비스를 유스케이스로 식별한다. 식별된 유스케이스에 대해 상호작용 내역을 기반으로 명칭을 부여하고 서비스 내역을 식별한다.
- 유스케이스 정의
유스케이스를 시작시키는 액터인 유발자와 시작되기 위한 사전조건, 주요 입력물, 그리고 종료된 후 시스템의 상태나 생성된 정보 등 사후조건과 주요 결과물을 식별하고 시스템의 주요 처리 내역을 정의한다.

3. 유스케이스 상세화

유스케이스 식별 결과를 기반으로 각 유스케이스의 사건 흐름을 작성하고 유스케이스를 정제해 유스케이스간 관계를 식별한다.

- 사건 흐름 정의
유스케이스가 무엇을 해야 하는가에 초점을 두어 액터와의 상호작용을 고려해 사건 흐름을 작성한다. 사건 흐름은 유스케이스의 기본 흐름과 대안 흐름 및 예외 흐름으로 구분해 작성한다. 사건 흐름 작성시 유스케이스에서 처리되어야 할 데이터에 대한 식별/정의를 함께 수행해 사건 흐름에 포함한다.
- 사건 흐름별 사전/사후 조건 정의
유스케이스 사건 흐름별로 사전조건과 사후조건을 식별한다. 사전조건은 사건 흐름이 시작할 수 있는 시스템의 상태이며, 사후조건은 사건 흐름이 종료되었음을 확인할 수 있는 시스템의 상태이다. 기본 흐름의 사전/사후 조건은 유스케이스의 사전/사후 조건이다.

■ 유스케이스 모델링 작업(1R31)

- 가변성 정의
유스케이스 식별 단위의 일반/상세 정도에 따라 해당 유스케이스가 다양한 경우에 가변적으로 적용될 수 있는 경우를 고려하여 가변성을 정의한다.
- 부가사항 정의
유스케이스는 기능적 요구사항에 초점을 두어 식별하므로 유스케이스와 관련이 있으면서 유스케이스의 기능이나 사건 흐름에서 고려되지 않은 비기능적 요구사항 및 연동 요구사항 등 기타 시스템이 만족해야 할 요구사항을 부가사항으로 정의한다.
- 시나리오 작성
해당 유스케이스의 사건 흐름이 복잡한 경우 이해도를 높이거나 개발된 시스템이 어떻게 사용되는지 명확히 하기 위해 유스케이스별로 시나리오를 작성한다.
- 관계 정의
액터, 유스케이스 식별 및 상세화 결과를 기반으로 액터간(일반화 관계), 액터와 유스케이스간, 유스케이스간(include, extend) 관계를 정의한다.

■ 작업 수행시 고려사항

■ 유스케이스 모델링의 의미

- 유스케이스 모델링은 개발할 시스템의 기능적 요구사항을 분석하는 것 외에도 시스템의 경계를 설정하고 시스템에서 제공해야 하는 기능을 도출하는 기준이 된다.
- 유스케이스는 시스템의 기능을 분석/설계하는데 기준이 되므로 적정한 유스케이스의 도출이 매우 중요하며, 다양한 관점에서 모델링과 정제를 반복해 누락되거나 잘못 도출되지 않도록 주의해야 한다.

■ 액터 식별시 고려사항

- 액터는 시스템과 상호작용하는 행위를 기반으로 식별하며, 액터가 시스템과 어떤 상호작용을 하는지가 명확히 식별되어야 한다.
- 액터는 조직, 타 시스템, 사람 등 물리적인 것이 아니라 시스템과 상호작용하는 논리적인 개념으로 액터가 시스템을 통해 달성하고자 하는 목적이 액터를 식별/정제하는 기준이다.
- 액터는 개발 대상 시스템의 경계를 판단하는데 중요한 요소이므로 무엇이 시스템이 제공해야 할 기능인지 아닌지를 명확히 해야 한다. 예를 들어 사용자의 인적 정보를 시스템에서 관리하지 않을 경우 인적 정보를 제공해줄 액터가 식별되어야 하며, 시스템은 액터로부터 전달받은 인적 정보를 처리하는 기능이 명시되어야 한다. 또한, 시스템의 접근을 통제하는 기능이나 보안과 관련된

■ 유스케이스 모델링 작업(1R31)

기능을 타 시스템이나 시설로부터 제공받을 경우 이를 액터로 식별해 시스템의 경계를 명확히 할 수 있다.

- 다음의 항목을 고려해 액터를 식별한다.
 - 시스템을 사용하고 있는(또는 사용할) 사용자는 누구인가?
 - 시스템에 정보를 입력하고 사용하고 삭제하는 사용자는 누구인가?
 - 시스템으로부터 정보를 얻는 사용자는 누구인가?
 - 어떤 부서나 조직에서 시스템을 사용하는가?
 - 시스템이 필요로 하는 정보를 소유한 타 시스템(또는 하드웨어, 소프트웨어)은 무엇인가?
 - 시스템이 정보를 제공하기로 협의된 타 시스템은 무엇인가?
- 액터의 명칭과 수행 역할을 명확하게 작성하여 액터간 혼동이 없도록 하며, 용어집과 일관성을 유지하도록 한다. 액터 식별 및 정제는 유스케이스 모델링을 수행하는 동안 지속적으로 수행한다.

■ 유스케이스 식별시 고려사항

- 유스케이스 식별을 수행하면서 상호보완적으로 액터와 유스케이스를 정제하고 액터와 유스케이스의 상호작용은 중복성이 적고 상호작용 내역을 구체적으로 파악할 수 있는 수준으로 작성한다.
- 유스케이스는 시스템의 기능, 시스템이 제공하는 서비스를 대상으로 하므로 시스템과 관련 없는 업무 내용은 유스케이스 식별시 고려하지 않는다.
- 유스케이스의 사건 흐름 중 대안 흐름은 기본 흐름에서 분기해 별도의 처리를 수행하나 다시 기본 흐름으로 복귀하는 흐름을 의미하며, 예외 흐름은 기본 흐름으로 복귀하지 않는 흐름을 의미한다.
- 유스케이스 식별시 유스케이스간 관계(include, extend 등) 도출은 고려하지 않고 수행하는 것이 유스케이스 명확화에 도움을 준다. 유스케이스간 관계를 고려하며 유스케이스를 도출할 경우 유스케이스 도출보다 관계에 초점을 두게 되어 정확한 유스케이스 도출이 어려워 질 수 있으며, 관계 설정에 계획 이상의 노력이 투입될 수 있기 때문이다.
- 유스케이스간 관계 설정은 액터에 대해 주요 유스케이스가 식별된 이후 유스케이스를 정제하며 설정/수정하는 것이 좋다.
- 개발 대상 시스템의 범위가 커서 식별된 유스케이스의 수가 많을 경우 비즈니스 정의서의 비즈니스 패키지를 기반으로 유스케이스를 패키징 한다.

유스케이스 모델링 작업(1R31)

- 하나의 유스케이스 다이어그램에 액터 또는 유스케이스의 수가 너무 많을 경우 유스케이스 모델 관리의 용이성을 높이기 위해 별도의 패키징을 통해 유스케이스간 계층 관계를 작성한다.
- 유스케이스 시나리오는 유스케이스가 시스템으로 구현되어 사용되는 모습으로써 유스케이스의 사건 흐름별로 시나리오를 작성한다. 단, 유스케이스의 사건 흐름이 명확하여 사건 흐름만으로 시스템의 동작을 이해할 수 있다면 작성하지 않아도 된다.
- 다음의 항목을 고려해 유스케이스를 식별한다.
 - 각 액터가 수행하는 업무는 무엇인가?
 - 추출한 유스케이스가 시스템에서 요구하는 모든 기능을 포괄하는가?
 - 액터와 상호작용하여 정보를 생성, 저장, 수정, 삭제하고 조회하는가?
 - 액터가 외부 상태 변화를 시스템에 알릴 필요가 있는가?
 - 시스템에서 발생한 상태 변화를 액터가 알아야 하는가?
 - 어떠한 유스케이스가 시스템을 지원하고 유지하는가?
- 식별된 유스케이스를 모두 종합한 것이 시스템의 범위이다. 각 유스케이스는 액터와의 상호작용에 의해 수행하고자 하는 것이 나타나야 하며, 개발할 시스템의 범위 및 능력의 기준이 되므로 상세하게 식별해야 한다.
- 액터간 관계, 액터와 유스케이스간의 상호작용, 유스케이스간의 관계는 시스템의 범위와 기능을 명확히 하는데 도움을 주나 불필요한 관계 설정은 시스템의 이해도를 낮출 수 있으므로 각 관계의 의미를 정확히 이해하고 도출해야 한다.
- 가변성은 하나의 유스케이스가 동일한 기능을 수행하지만 다양한 경우에 차별적/차등적으로 적용되는 경우를 식별하는 것으로써 다음과 같은 항목을 고려하여 식별한다.
 - 내부에 다양한 연산 로직(logic)이 존재하는가?
 - 내부에 다양한 정보 속성(attribute)이 존재하는가?
 - 내부에 다양한 비즈니스 흐름(workflow)이 존재하는가?
 - 기타 변화할 수 있는 다양한 경우가 존재하는가?
- 부가사항 중 연동 요구사항은 액터가 타 시스템인 경우 연동 내역에 대한 식별을 수행하고 유스케이스와의 상호작용 내역을 식별해 작성할 수 있다. 연동 내역은 통신 프로토콜, 연동 기술, 연동 정보 등이며, 상세화는 이후 작업에서 고려하고 연동 내역을 누락 없이 완전하게 식별하는데 초점을 둔다.

■ 유스케이스 모델링 작업(1R31)

■ 유스케이스 모델링 결과와 이후 작업과의 관계

- 액터와 유스케이스간의 상호작용은 사용자 인터페이스 또는 타 시스템과의 연동 등으로 도출되어 연동 소요 분석(1R32), 사용자 인터페이스 정의(1R33)에서 연동 소요 또는 화면/보고서로 유도되며, 이후 설계(2D) 단계에서 UI 컴포넌트, 사용자 화면, 연동 항목 등으로 구체화 된다.
- 유스케이스는 이후 시스템의 기능 분석 및 설계의 기본으로 클래스 모델링(1R34) 작업에서 클래스로 유도되며, 컴포넌트 식별(2D11) 작업에서 컴포넌트 식별을 위한 기준으로 활용된다.
- 유스케이스 모델링 완료 후 그 결과를 요구사항 추적표에 반영하고 모든 요구사항에 대해 유스케이스가 식별되었는지를 확인해야 한다.

■ 유스케이스 모델링시 표기법

- 유스케이스 모델링을 위한 표기법은 UML의 유스케이스 다이어그램 표기법을 사용하며, 사업의 특성을 고려해 UML의 구조를 준용해 표기법을 확장할 수 있다.

■ 유스케이스 모델링 구성요소간 관계

- 액터는 하나 이상의 유스케이스와 관계를 가지며, 유스케이스는 없거나 하나 이상의 액터와 관계를 가진다. 액터와 유스케이스간 관계가 없는 유스케이스는 include, extend 관계로 연결된 유스케이스이다. 유스케이스와 액터간의 관계는 방향성이 있다.
- 유스케이스간 관계는 include, extend 관계만 존재하며, 방향성이 있다. 유스케이스간 관계는 관계 설정을 위한 제약조건이 존재할 수 있다.
- 유스케이스는 없거나 하나 이상의 유스케이스 시나리오를 포함하며, 유스케이스 시나리오는 반드시 하나의 유스케이스에 포함된다.
- 유스케이스는 하나의 기본 흐름을 가지며, 없거나 하나 이상의 대안 흐름 또는 예외 흐름을 포함한다. 유스케이스 대안 흐름과 예외 흐름은 하나의 유스케이스에만 포함된다.
- 유스케이스 가변성은 이후 컴포넌트의 가변성으로 식별되거나 별도 기능으로 분리된다.

■ 유스케이스 모델링 작업(1R31)

■ [서식] 유스케이스 정의서(1R31a)

1. 액터 목록

액터명	액터 설명	관련 원천

2. (유스케이스 다이어그램명)

2.1. 유스케이스 다이어그램

2.2. (유스케이스명)

2.2.1. 개요

2.2.2. 관련 액터

액터명	입력 정보	출력 정보

2.2.3. 사건 흐름

2.2.4. 가변성

2.2.5. 부가사항

2.2.6. 시나리오

■ 유스케이스 모델링 작업(1R31)

■ [항목 설명] 유스케이스 정의서(1R31a)

1. 액터 목록

개발 대상 시스템의 액터 목록을 작성한다.

- 액터명
액터의 명칭을 기재한다.
- 액터 설명
액터의 주요 임무와 시스템과의 상호작용을 설명한다.
- 관련 원천
액터로 묶인 내/외부 조직, 담당자 또는 시스템 등의 명칭을 기술한다.

2. (유스케이스 다이어그램명)

2.1. 유스케이스 다이어그램

액터와 유스케이스들의 관계를 유스케이스 다이어그램으로 작성한다.

2.2. (유스케이스명)

유스케이스 다이어그램에 포함된 유스케이스별로 작성한다.

2.2.1. 개요

유스케이스가 제공하는 시스템의 범위와 기능에 초점을 두어 유스케이스의 개요를 작성한다.

2.2.2. 관련 액터

해당 유스케이스와 상호작용하는 액터별로 유스케이스와의 상호작용 내역인 입출력 정보를 기술한다.

- 액터명
해당 유스케이스와 상호작용하는 액터의 명칭을 기재한다.
- 입력 정보
액터가 유스케이스로 입력하는 정보(또는 정보 명칭)를 기재한다.
- 출력 정보
유스케이스가 액터에게 출력하는 정보(또는 정보 명칭)를 기재한다.

■ 유스케이스 모델링 작업(1R31)

2.2.3. 사건 흐름

해당 유스케이스의 사건 흐름을 기본 흐름, 대안 흐름, 예외 흐름으로 구분해 작성한다.

- 기본 흐름
해당 유스케이스가 일반적으로 수행하게 될 처리 흐름을 기술한다.
- 대안 흐름
기본 흐름을 진행하다가 조건에 따른 분기로 인해 다른 처리 흐름을 수행하는 경우 이를 기술한다.
※ 대안 흐름이 종료되면 기본 흐름의 특정 위치로 복귀해 나머지 기본 흐름을 수행한다.
- 예외 흐름
기본 흐름을 진행하다가 조건에 따른 분기로 인해 예외적인 처리 흐름을 수행하는 경우 이를 기술한다.
※ 예외 흐름은 기본 흐름으로 복귀되지 않고 종료된다.

2.2.4. 가변성

유스케이스 수행에 가변성이 있는 경우 기술한다.

2.2.5. 부가사항

성능, 하드웨어 사양, 보안 등과 같은 비기능적 요구사항 및 타 시스템과의 연동 요구사항 등 유스케이스와 관련 있는 부가사항을 기술한다.

2.2.6. 시나리오

유스케이스가 시스템으로 개발되어 수행될 때를 가정하여 실제 사용자 및 관련 정보, 처리 내역을 시나리오로 기술한다.

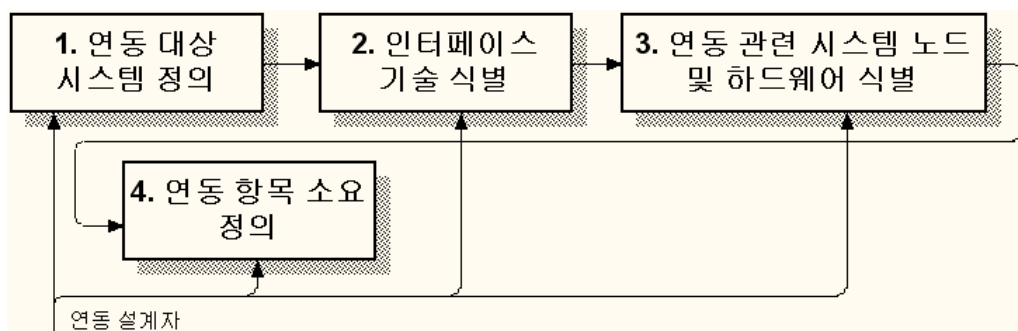
연동 소요 분석 작업(1R32)

개요

개발 대상 시스템이 타 시스템과의 연동 소요를 가질 때 연동에 대한 대상을 정의하고 이를 구현하기 위한 환경과 시스템, 기술, 범위, 방법 등을 식별하여 가능성과 실현 방안을 탐색하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 비즈니스 정의서 요구사항 정의서, (부록)요구사항 추적표 시스템 아키텍처 정의서 유스케이스 정의서
산출물	<ul style="list-style-type: none"> 연동 소요 정의서, (부록)연동 항목 소요서(IER)
참여자	<ul style="list-style-type: none"> 연동 설계자 <p>연동 요구사항 및 시스템 아키텍처를 바탕으로 연동 대상과 제반 환경, 연동 항목, 구현 기술을 도출한다.</p>
관련기법	해당 사항 없음
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 연동 대상 시스템 정의

연동에 참여하게 될 개발 대상 시스템과 타 시스템을 정의한다.

연동 소요 분석 작업(1R32)

- 연동 대상 시스템 정의
개발 대상 시스템(또는 서브시스템)과 연동 대상인 타 시스템을 식별하여 정의한다. 개발 대상 시스템에 별도의 서브시스템이 존재하지 않거나 특정 서브시스템에 관계없이 전체 시스템에서 연동이 요구된다면 최상위 시스템만을 정의한다.
- 시스템간 인터페이스 식별
대상이 결정되면 개발 대상 시스템과 타 시스템간의 인터페이스를 유일하게 식별할 수 있도록 인터페이스명을 부여한다.

2. 인터페이스 기술 식별

각 인터페이스에 대해 구현을 위해 사용되는 기술 환경에 대해 정의한다.

- 인터페이스 기술 식별
일반적으로 널리 알려진 기술이나 DITA(Defence Information Technical stAndard)를 기준으로 인터페이스 기술을 식별하고 각 기술을 적용하기 위한 전제조건 또는 제약사항을 식별한다.
- 인터페이스 기술 구성 식별
인터페이스 기술을 명확히 정의하기 위하여 이를 구성하는 요소와 이들 간의 흐름을 정의한다.

3. 연동 관련 시스템 노드 및 하드웨어 식별

개발 대상 시스템 및 타 시스템에서 기술 구성 요소가 탑재되는 하드웨어를 식별한다.

- 시스템 노드 및 하드웨어 식별
개발 대상 시스템 및 타 시스템의 하드웨어가 운용되는 실질적 노드(장소)를 식별하여 실제 운용 환경과 하드웨어의 관리 주체를 확인한다.

4. 연동 항목 소요 정의

각 연동 항목에 대해 주요 데이터 항목과 기타 특성, 연동 방법, 송신 또는 수신하는 시스템을 정의한다.

- 연동 항목 소요 정의
각 연동 항목에 대해 연동 항목 소요서(IER : Information Exchange Requirement)를 작성하여 정의한다.

연동 소요 분석 작업(1R32)

작업 수행시 고려사항

■ 연동 소요 분석 작업의 수행 여부

- 연동 소요 분석 작업은 개발 대상 시스템이 연동 소요를 가질 경우에 한해 반드시 실시해야 하며 연동 소요가 없는 경우에는 생략 가능하다.

■ 연동 항목 소요서(IER) 관리 및 갱신

- 연동 항목 소요서(IER)는 시스템 개발 단계 이전에 소요 제기 또는 개발 계획 단계에서부터 작성될 수 있다. 그러나 시스템이 개발 단계와 운용 단계를 거치면서 변경되는 연동 소요에 대해 지속적으로 변경하여 관리될 필요가 있다. 따라서 연동 소요 정의서와 연동 항목 소요서(IER)를 분리하여 변경되는 부분에 대해 갱신하여 관리해야 한다.

■ 인터페이스 기술 구성 식별시 고려사항

- 개발 대상 시스템의 인터페이스 기술 구성 요소가 물리적으로는 타 시스템 하드웨어에 별도의 인스턴스로 분리되어 탑재되거나 반대로 타 시스템의 요소가 개발 대상 시스템의 하드웨어에 탑재되는 경우가 존재할 수 있으므로 이에 대한 식별을 수행한다.
- 만일 타 시스템의 하드웨어에 인터페이스 기술 구성 요소가 신규로 탑재되어야 한다면 대상 하드웨어가 적절한 사양과 환경을 갖추고 있는지에 대한 조사가 필요하다.
- 대상 하드웨어 사양과 운영체제 종류 및 버전, 가상 머신 버전 등을 파악하여 적합한 환경을 갖추고 있는지에 대한 판단을 실시하고 대안을 제시한다.

■ 연동 관련 시스템 노드 및 하드웨어 식별시 고려사항

- 필요시 시스템 노드의 네트워크 대역폭과 속도를 점검하여 연결성을 판단하고 대안을 제시한다.

■ 인터페이스 기술의 사전 정의시의 산출물 작성

- 패키지 소프트웨어나 상용 프레임워크의 도입, 기존 시스템의 커스터마이징, 타 시스템의 연동 모듈 제공 등과 같이 연동하기 위한 기술 요건이 이미 정의되었다면 이를 바탕으로 기술하며, 필요시 산출물 서식을 추가하여 작성할 수 있다.
- 해당 기술 적용에 필요한 별도의 분석 단계 산출물(예 : EAI 표준화 정의서 등)이 존재한다면 부록으로 첨부한다.

연동 소요 분석 작업(1R32)

■ 프로토타이핑 기법 활용

- 필요시 사용자와 요구 내용의 이해 및 검증, 연동 성능 평가 등의 목적으로 프로토타입을 제작하여 활용할 수 있다.

■ 연동 소요 분석 결과와 이후 작업과의 관계

- 연동 소요 분석 작업은 연동 설계 작업(2D24)과 작업 시점을 고려하여 분리하였으나 관련성이 높으므로 2개 작업을 반복 수행하면서 연동에 관한 분석 및 설계 결과의 완전성을 높여야 한다.
- 연동 항목 소요서(IER)는 연동 설계 작업의 인터페이스 데이터 목록 및 인터페이스 데이터 상세로 유도되어 구체화된다.

연동 소요 분석 작업(1R32)

[서식] 연동 소요 정의서(1R32a)

1. 개요

1.1. 연동 개념도

1.2. 개발 대상 시스템 정의

시스템명	시스템 수준	상위 시스템명	시스템 노드

1.3. 타 시스템 정의

타 시스템명	설 명	시스템 노드

1.4. 연동 대상 시스템 노드

시스템 노드명	설 명	네트워크 특성

연동 소요 분석 작업(1R32)

2. 연동 소요 정의

2.1. 인터페이스 정의

인터페이스명	설 명	출발 시스템	도착 시스템

3. 연동 환경 분석

3.1. 인터페이스 기술

인터페이스명	인터페이스 기술명	설 명	제약사항

연동 소요 분석 작업(1R32)

3.2. 인터페이스 기술 구성 요소

인터페이스 명	인터페이스 기술	구성 요소		설 명	관련 하드웨어명
		구분	요소명		

3.3. 인터페이스 구성 요소 흐름

인터페이스명	인터페이스 기술	구성 요소 흐름명	설 명	출발 구성 요소	도착 구성 요소

3.4. 인터페이스 하드웨어

하드웨어명	관련 네트워크명	관련 운용 환경명	관련 시스템 노드

연동 소요 분석 작업(1R32)

■ [항목 설명] 연동 소요 정의서(1R32a)

1. 개요

1.1. 연동 개념도

연동 구성을 가시적으로 확인할 수 있도록 연동에 참여하는 시스템, 시스템 노드, 운용 환경, 인터페이스명, 기술 요소, 하드웨어 등으로 구성되는 개념도를 작성하고 세부적인 서술이 필요할 경우 이를 포함한다. 만일 개발 대상 시스템이 다수의 복잡한 인터페이스를 가진다면 각각의 인터페이스에 대해 연동 개념도를 별도로 작성할 수 있다.

개념도는 특정한 다이어그램의 형식이나 또는 표기법을 따르는 것이 아니므로 필요시 정적/동적인 모습을 표현하도록 한다.

1.2. 개발 대상 시스템 정의*

- 시스템명
시스템을 대표하는 명칭을 기재하되, 연동에 참여하는 시스템 또는 서브시스템명을 식별한다. 특정 서브시스템에 관계없이 전체 시스템에서 연동이 요구된다면 최상위 시스템만을 기재한다.
- 시스템 수준
시스템간의 상하관계를 나타내며, 최상위 수준을 “1”로 표시하고 하위 수준은 “점_숫자”의 형태로 기재한다. (예 : 1.1, 1.1.21, 1.1.21.1 등)
- 상위 시스템명
시스템을 포함하는 상위 시스템이 있을 경우 이를 대표하는 명칭을 기재한다.
- 시스템 노드
시스템이 운용되는 실질적 노드(장소)를 기재한다. (예 : 00부대 전산실)

1.3. 타 시스템 정의

- 타 시스템명
개발 대상 시스템과의 연동에 참여하는 시스템을 대표하는 명칭을 기재한다.
- 설명
시스템에 대한 설명을 기술한다.

1.4. 연동 관련 시스템 노드*

- 시스템 노드명
개발 대상 시스템 및 타 시스템이 운용되는 실질적 노드(장소)를 기재한다.
(예 : 00부대 전산실)

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

연동 소요 분석 작업(1R32)

- 설명
상위 부대, 지리적 위치 등 시스템 노드에 대한 설명을 기술한다.
- 네트워크 특성
속도, 대역폭, 유의사항 등 시스템 노드의 네트워크 특성을 기술한다.

2. 연동 소요 정의

2.1. 인터페이스 정의*

- 인터페이스명
시스템간의 인터페이스를 대표하는 명칭을 기재한다.
- 설명
시스템 인터페이스에 대한 설명과 인터페이스 방향에 대한 추가적인 정보를 기술한다.(시스템간 인터페이스는 기본적으로 양방향일 의미한다)
- 출발 시스템
인터페이스가 단방향일 경우 연동되는 정보가 출발하는 시스템명을 기재하고 양방향일 경우 생략한다.
- 도착 시스템
인터페이스가 단방향일 경우 연동되는 정보가 도착하는 시스템명을 기재하고 양방향일 경우 생략한다.

3. 연동 환경 분석

3.1. 인터페이스 기술

- 인터페이스명
시스템간의 인터페이스를 대표하는 명칭을 기재한다.
- 인터페이스 기술명
인터페이스를 구현하는 표준 기술명 또는 일반적으로 널리 알려진 통상적인 명칭(예 : 웹 서비스, EAI, 소켓, DB링크 등)을 기재한다.
- 설명
인터페이스 기술에 대한 설명을 기술한다.
- 제약사항
해당 인터페이스 기술을 활용할 경우의 제약사항 또는 사전 조건을 기술한다.

3.2. 인터페이스 기술 구성 요소

- 인터페이스명
시스템간의 인터페이스를 대표하는 명칭을 기재한다.

연동 소요 분석 작업(1R32)

- 인터페이스 기술
인터페이스를 구현하는 표준 기술명 또는 일반적으로 널리 알려진 통상적인 명칭을 기재한다.
- 구성 요소명
인터페이스 기술을 구현한 구성 요소(소프트웨어명 또는 컴포넌트명 등)를 기술한다.
- 설명
기술 구성 요소에 대한 설명을 기술한다.
- 관련 하드웨어명
구성 요소가 탑재되는 하드웨어명을 기재한다.

3.3. 인터페이스 구성 요소 흐름

인터페이스 구성 요소간의 흐름을 나타내되, 표 형식 또는 정적/동적 다이어그램, 개념도, 혹은 이들을 적절히 조합하여 작성할 수 있다.

- 인터페이스명
시스템간의 인터페이스를 대표하는 명칭을 기재한다.
- 인터페이스 기술
인터페이스를 구현하는 표준 기술명 또는 일반적으로 널리 알려진 통상적인 명칭을 기재한다.
- 구성 요소 흐름명
인터페이스 기술 구성 요소간의 정보 흐름을 대표하는 명칭을 기재한다.
- 설명
구성 요소 흐름에 대한 설명과 흐름 방향에 대한 추가적인 정보를 기술한다.
(흐름은 기본적으로 양방향성을 의미한다)
- 출발 구성 요소
흐름이 단방향일 경우 흐름이 출발하는 구성 요소명을 기재하고 양방향일 경우 생략한다.
- 도착 구성 요소
흐름이 단방향일 경우 흐름이 도착하는 구성 요소명을 기재하고 양방향일 경우 생략한다.

3.4. 인터페이스 하드웨어

- 하드웨어명
인터페이스 기술 구성 요소가 탑재되는 하드웨어명을 기재한다.

연동 소요 분석 작업(1R32)

- 관련 네트워크명
하드웨어가 연결되어 있는 네트워크명을 기재한다.
- 관련 운용 환경명
하드웨어가 배치되는 운용 환경명을 기재한다.
- 관련 시스템 노드
운용 환경과 관련 있는 시스템 노드를 기재한다.

연동 소요 분석 작업(1R32)

[서식] 부록 - 연동 항목 소요서(IER)

1. (인터페이스명)

순번	연동 항목명	송신 시스템	수신 시스템	주요 데이터 항목	설명	용도	주기	보안등급	연동방법	비고

연동 소요 분석 작업(1R32)

■ [항목 설명] 부록 - 연동 항목 소요서(IER)

1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭으로써 연동 소요 정의서에서 식별된 명칭을 기재한다.

- 순번
일련번호 또는 식별할 수 있는 식별자를 기재한다.
- 연동 항목명
연동되는 정보의 묶음을 식별할 수 있는 명칭을 기재한다.
- 송신 시스템
연동 정보를 송신하는 시스템을 대표하는 명칭이나 식별할 수 있는 범위 내에서 약어를 기재한다.
- 수신 시스템
연동 정보를 수신하는 시스템을 대표하는 명칭이나 식별할 수 있는 범위 내에서 약어를 기재한다.
- 주요 데이터 항목
연동 항목을 구성하는 주요 논리적 데이터 항목을 기재하되, 특정 플랫폼에 의존적이거나 반복적인 부가 정보(예 : 입력일시)는 생략이 가능하다.
- 설명
연동 항목에 대한 설명과 연동 항목이 생성되는 업무활동, 조건 또는 제약사항 등에 대해서 기술한다. 또한 연동 항목을 송신하는 시스템과 최초로 생성하는 시스템이 다를 경우, 연동 항목을 생성하는 시스템의 명칭을 추가로 기술한다. 반대로 연동 항목을 수신하는 시스템과 최종 사용하는 시스템이 다를 경우도 기술한다.
(예 : A시스템이 C시스템에게 정보를 전달하고 B시스템은 중간에서 중계를 할 경우에 C시스템 입장에서 보면 송신 시스템은 B시스템이 되지만 정보를 최초로 생성하는 시스템은 A시스템이다.)
- 용도
연동 항목이 필요한 비즈니스 활동 정보를 기술한다.
- 주기
연동이 발생하는 주기를 기재하되, 모드(예 : 전시, 평시)별로 상이할 경우 이를 구분하고 특정 시기(예 : 1월 1일, 24:00, 06:00)가 요구될 경우도 식별하여 기재한다.
- 보안등급
'군사보안업무훈령'에 정의된 보안등급을 기재한다.

연동 소요 분석 작업(1R32)

- 연동방법
연동을 구현하는 표준 기술명 또는 일반적으로 널리 알려진 통상적인 명칭으로 써, 연동 소요 정의서를 통해 식별된 인터페이스 기술명을 기재한다.
- 비고
기타 추가적인 정보를 기술한다.

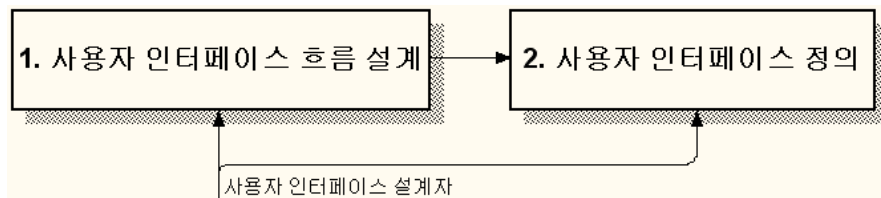
■ 사용자 인터페이스 정의 작업(1R33)

■ 개 요

개발 대상 시스템에 대한 요구사항을 확인 및 정제하고 사용자 인터페이스의 편의성을 검증받기 위해, 모든 화면의 레이아웃 및 구성항목에 대하여 정의하고, 검증 및 보완하는 작업이다. 또한, 시스템과 사용자 간의 인터페이스를 담당하는 사용자 화면간의 제어 흐름을 정의한다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 요구사항 정의서, (부록)요구사항 추적표 유스케이스 정의서
산출물	<ul style="list-style-type: none"> 사용자 인터페이스 정의서
참여자	<ul style="list-style-type: none"> 사용자 인터페이스 설계자 사용자와 시스템 간의 인터페이스를 담당하는 화면의 레이아웃 및 구성항목에 대하여 정의한다. 사용자 정의된 사용자 인터페이스에 대한 검토를 실시한다.
관련기법	<ul style="list-style-type: none"> UML

■ 절차 흐름도



■ 절차 설명

1. 사용자 인터페이스 흐름 설계

유스케이스 모델을 바탕으로 시스템과 액터간의 인터페이스를 담당하는 화면을 도출하고, 도출된 화면간의 제어 흐름을 정의한다.

- 사용자 인터페이스 흐름 설계

개발 대상 시스템의 전체적인 화면 흐름을 한눈에 볼 수 있도록 상위 사용자 화면을 중심으로 하위에 위치하는 여러 사용자 화면과의 관계를 메뉴 흐름도 또는 표 형태 등으로 나타낸다. 유스케이스 설명을 바탕으로 사용자 화면간의 제어 흐름을 액티비티 다이어그램 형태로 정의한다.

■ 사용자 인터페이스 정의 작업(1R33)

2. 사용자 인터페이스 정의

모든 화면 및 보고서 등 사용자 인터페이스를 정의한다.

- 사용자 인터페이스 화면 정의

유스케이스 모델과 사용자 인터페이스 흐름도를 바탕으로 개발 대상 시스템의 모든 화면 목록을 작성하고, 각 화면에 대해서 레이아웃 및 구성항목을 포함하는 최종적인 화면을 정의한다. 여러 화면에서 공통적으로 사용되는 화면에 대해서는 별도로 분류하여 기술하는 것을 검토한다.

- 보고서 정의

보고서 출력과 관련된 요구사항이 있는 경우 화면과 마찬가지로 보고서의 레이아웃 및 구성항목을 작성하며 화면과 구성항목들이 유사한 경우 참조 처리하도록 한다.

■ 작업 수행시 고려사항

■ 사용자 인터페이스 흐름 설계시 고려사항

- 사용자의 시스템 사용 환경과 사용 능력, 사용자의 수 및 사용빈도 그리고 표준 지침에서 정의한 사용자 인터페이스 표준 등을 고려하여 사용자 인터페이스에 대한 흐름을 설계한다.
- 기간 선택, 부대 선택 등과 같이 반복적이고, 공통적으로 사용되는 부화면에 대한 흐름은 생략 가능하다.

■ 사용자 인터페이스 정의시 고려사항

- 사용자 인터페이스는 사용자의 기능적 요구사항을 충족시키는 동시에 기술적 요구사항을 만족해야 하므로 사용자 및 사용자 인터페이스 설계자가 함께 화면을 설계한다.
- 사용자 인터페이스는 사용자가 직관적으로 기능을 이해할 수 있도록 구성되어야 하며, 사용자 인터페이스와 관련된 표준이 있다면 표준을 준수하여 작성하도록 한다. 또한, 사용자 인터페이스를 개발하는 도구를 이용한다면 화면을 작성할 때 해당 도구를 사용한다.
- 시스템 기능 및 보안 요구사항에 따라 화면별로 사용자의 접근 및 사용 권한이 다른 경우 이를 정의한다.
- 분석단계에 모든 사용자 인터페이스에 대한 정의가 이루어지고, 그에 대한 사용자 검토가 시행되어야 하므로 단계별 일정 계획수립 시 충분한 시간이 배정되어야 한다.

■ 사용자 인터페이스 정의 작업(1R33)

- 사용자 화면은 드로잉, 비트맵, 실행 모듈 등의 방식을 이용하여 실제 화면으로 사용할 수 있는 화면 레이아웃으로 정의하며, 해당 화면의 세부 항목속성은 Menu, Tool Bar, Control Box(Text Box, Push Button, Radio Button...) 등으로 정의한다.
- 사용자 인터페이스 정의 완료 후 그 결과를 요구사항 추적표에 반영하여 유스케이스와 사용자 인터페이스간의 관계 확인이 가능하도록 한다.

■ 사용자 인터페이스 정의 작업(1R33)

■ [서식] 사용자 인터페이스 정의서(1R33a)

1. 메뉴 구성

시스템	Level 1	Level 2	Level 3

2. (기능 이름)

2.1. 화면 목록

화면ID	화면명	화면 설명	보안 등급	액터명	권한

2.2. 화면 흐름도

2.3. 화면 정의

2.3.1. (화면명(화면ID))

2.3.1.1. 설명

2.3.1.2. 레이아웃

2.3.1.3. 항목 설명

구분	항목명	항목 속성	설명

2.4. 보고서 정의

2.4.1. (보고서명(보고서ID))

2.4.1.1. 설명

■ 사용자 인터페이스 정의 작업(1R33)

2.4.1.2. 레이아웃

2.4.1.3. 항목 설명

항목명	설명

■ 사용자 인터페이스 정의 작업(1R33)

■ [항목 설명] 사용자 인터페이스 정의서(1R33a)

1. 메뉴 구성

개발 대상 시스템의 사용자 인터페이스를 구성하고 있는 메뉴를 계층 형태로 나타낸다. 최상위 시스템을 중심으로 하위에 위치하고 있는 여러 메뉴 및 화면과의 관계를 나타내어 전체적인 화면의 흐름을 한눈에 볼 수 있도록 한다. 필요시 표 형태가 아닌, 트리 형태의 구조도로 작성 가능하다.

- 시스템
최상위 시스템 또는 서브시스템을 대표하는 명칭을 기재한다.
- Level 1~3
하위 메뉴 명칭을 기재한다. 개발 대상 시스템에 맞도록 레벨 단계를 조정하여 작성한다.

2. (기능 이름)

메뉴에 연결되는 기능의 이름을 기재하고, 기능별로 화면 목록 및 흐름도와 화면 및 보고서 정의를 작성한다.

2.1. 화면 목록

기능내 주화면과 부화면을 포함하는 모든 화면에 대한 목록을 작성한다.

- 화면 ID
화면의 식별자를 기재한다.
- 화면명
화면의 이름을 기재한다.
- 화면 설명
화면에 대한 간단한 설명을 기술한다.
- 보안등급
해당 화면의 보안 등급을 '군사보안업무훈령'에 따라 분류하여 기재한다.
- 액터명
해당 화면에 관련된 액터명을 기재하되 유스케이스 정의서에서 정의된 액터명으로 기재한다.
- 권한
관련된 액터에 따라 접근 및 사용 권한이 다를 경우 이를 명시한다. 권한에 대한 정의는 기능 단위로 세세하게 기술할 수 있으며, 또는 '전체 권한' 등으로 간략한 기술이 가능하다. 필요시 계층적 권한 등 세부적 권한 체계를 별도로 정의한다.

■ 사용자 인터페이스 정의 작업(1R33)

2.2. 화면 흐름도

유스케이스 설명에 따라 시스템의 구현 및 시스템의 재사용을 고려하여 사용자 화면간의 제어 흐름을 액티비티 다이어그램으로 나타낸다. 주요한 화면의 흐름을 위주로 작성하며 자명한 흐름은 생략 가능하다.

2.3. 화면 정의

2.3.1. (화면명(화면ID))

화면 목록에 있는 화면명과 화면ID를 기재한다.

2.3.1.1. 설명

화면의 용도 및 기능에 대한 설명을 간략하게 기술한다.

2.3.1.2. 레이아웃

화면 형식을 드로잉, 비트맵 등의 방식을 이용하여 도식화한다.

2.3.1.3. 항목 설명

- 구분
항목을 구분하는 것이 필요할 경우 구분자를 기재한다.
- 항목명
화면의 각 구성 개체들의 이름을 기재한다.
- 항목속성
항목의 속성을 기재한다. 항목 속성으로는 Menu, Tool Bar, Control Box(Text Box, Push Button, Radio Button...) 등을 기재한다.
- 설명
항목의 용도 및 기능, 동작 방식, 제약사항 등에 대해 간략하게 기술한다. 다른 팝업 화면이나 보고서를 호출하는 버튼의 경우 해당 화면명(화면ID)나 보고서명(보고서ID)을 함께 기술한다.

2.4. 보고서 정의

해당 기능에 보고서가 존재하는 경우 해당 보고서 서식과 관련된 내용에 대해 기술한다.

2.4.1. (보고서명(보고서ID))

2.4.1.1. 설명

보고서의 용도 및 보안 등급 등에 대한 설명을 간략하게 기술한다.

사용자 인터페이스 정의 작업(1R33)

2.4.1.2. 레이아웃

보고서 서식을 드로잉, 비트맵 등의 방식을 이용하여 도식화한다.

2.4.1.3. 항목 설명

- 항목명
보고서 항목의 이름을 기재한다.
- 설명
보고서 항목에 대하여 설명을 기술한다.

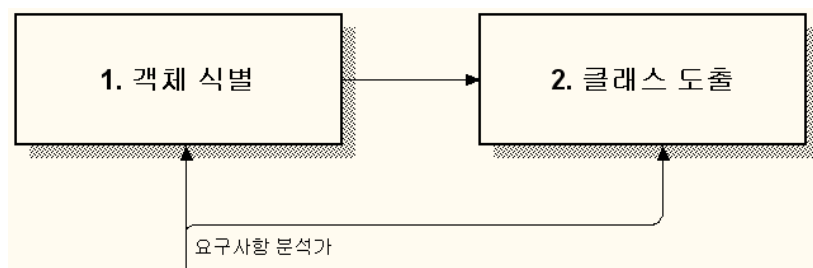
클래스 모델링 작업(1R34)

개 요

비즈니스 정의 결과와 유스케이스 모델링 결과를 기반으로 유스케이스 실현에 필요한 객체를 도출하고 객체간의 상호작용을 분석한다. 또한, 분석된 객체로부터 클래스를 도출하고 클래스의 속성 및 오퍼레이션과 클래스간의 관계를 분석하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 비즈니스 정의서 유스케이스 정의서
산출물	<ul style="list-style-type: none"> 클래스 정의서
참여자	<ul style="list-style-type: none"> 요구사항 분석가 객체를 식별하고 클래스를 도출한다.
관련기법	<ul style="list-style-type: none"> UML

절차 흐름도



※ 클래스 모델링의 각 절차는 한 번씩만 수행하는 것이 아니라 절차 흐름을 계속 반복하면서 수행한다.

절차 설명

1. 객체 식별

유스케이스 구현을 위한 객체를 식별한다.

• 객체 식별

유스케이스별로 유스케이스를 정보시스템으로 실현하기 위해 필요한 객체를 식별하고 객체간의 상호작용을 분석해 정보시스템의 구동 행위를 분석한다. 객체 간 상호작용 분석은 각 객체의 책임을 명확히 하고 객체를 정련하기 위함이다.

■ 클래스 모델링 작업(1R34)

객체는 유스케이스의 사건 흐름(또는 시나리오)을 기반으로 식별하며, 객체간 상호작용 분석을 통해 정보시스템으로 실현시 필요한 신규 객체를 식별하여 모호한 사건 흐름(또는 시나리오)의 내용을 보완할 수 있다.

2. 클래스 도출

식별된 객체를 기반으로 클래스를 도출한다. 클래스는 유사한 구조, 행위 및 관계를 갖는 객체의 추상화된 형태이다.

- 클래스 분류
식별된 객체를 추상화해 클래스를 분류한다.
- 클래스 정의
클래스의 의미를 명확히 하기 위해 스테레오 타입을 적용한다. 객체간 상호작용에서 식별된 메시지를 기반으로 클래스의 속성과 오퍼레이션을 정의하고 클래스간의 관계를 명시적으로 표현한다.

■ 작업 수행 지침

■ 클래스 모델링의 의미

- 유스케이스 모델링이 어떤 서비스가 액터에게 제공해야 하는가에 초점을 두었다면, 클래스 모델링은 어떤 서비스가 정보시스템에서 실현되어야 하는가에 초점을 둔다.
- 클래스 모델링은 개발할 시스템의 내부 구조를 분석하여 시스템이 처리하는 기능을 구체화하고 이후 컴포넌트를 설계하는 기준이 된다.

■ 객체 식별시 고려사항

- 객체는 정보시스템이 구동하는 행위 관점에서 식별한다. 따라서 액터로부터 정보를 받거나 전달하는 객체, 서비스를 처리하는 객체, 정보시스템에서 데이터를 제어하는 객체 등 다양한 관점에서 식별한다.
- 객체간 상호작용 분석은 통해 유스케이스의 사건 흐름(또는 시나리오)을 정보시스템 내부의 메시지를 식별하고 제어 순서를 정의하는 것에 초점을 둔다.

■ 클래스 식별시 고려사항

- 클래스의 스테레오 타입은 클래스의 의미를 명확히 하는데 초점을 두나 투입 노력 대비 사용 빈도는 낮으므로 클래스의 스테레오 타입이 지속적으로 활용되지 않을 경우 제시된 엔티티, 컨트롤, 바운더리 클래스를 기본으로 사용하며, 필요시 소프트웨어 아키텍처를 고려해 스테레오 타입을 확장한다.

클래스 모델링 작업(1R34)

- 일반적으로 클래스의 스테레오 타입은 다음의 3가지에서 선택하며, 클래스의 의미를 확장할 필요가 있을 경우 자체 스테레오 타입을 정의한다.

스테레오 타입명	설 명
엔티티 클래스 (Entity Class)	도메인의 정보를 저장하고 있거나, 정보를 제공하는 역할을 수행하는 클래스로 프로그램이 종료된 후에도 영구적으로 저장되는 클래스이다.
컨트롤 클래스 (Control Class)	처리 흐름을 제어(또는 조정)하거나 트랜잭션을 관리하는데 사용하는 클래스이다.
바운더리 클래스 (Boundary Class)	시스템 외부의 액터와 메시지를 주고받는 클래스로 시스템 외부와의 상호작용을 책임진다.

- 엔티티 클래스의 경우 비즈니스 모델링 작업의 비즈니스 개념 등 대부분의 엔티티 클래스는 선행 작업을 통해 이미 식별되어 있을 수 있고 유스케이스 모델링, 객체 식별 및 클래스 도출 과정에서 시스템 구동에 필요한 엔티티 클래스를 추가 한다.
- 컨트롤 클래스는 시스템이 수행하는 모든 제어를 포함하는 것은 아니며, 액터가 시스템의 제어를 수행하는 경우 컨트롤 클래스는 생략될 수 있으며(이 경우 바운더리 클래스가 제어를 담당), 반대로 제어가 너무 복잡한 경우 여러 개의 컨트롤 클래스가 제어를 분산해 담당할 수 있다.
- 바운더리 클래스는 사용자 인터페이스, 시스템 인터페이스 및 장치 인터페이스를 중심으로 도출한다. 사용자 인터페이스와 관계된 바운더리 클래스는 사용자 인터페이스 정의 작업에서 정의된 사용자 화면에 대응되도록 도출하며, 시스템 및 장치 인터페이스와 관련된 바운더리 클래스는 연동 소요 분석 작업의 타 시스템과의 연동 내역이나 장치에 종속적인 처리를 중심으로 도출한다.
- 객체간 상호작용의 각 메시지는 메시지를 수신하는 객체의 클래스 오퍼레이션으로 정의될 수 있으며, 오퍼레이션을 수행하기 위해 필요한 속성이나 엔티티 클래스에서 지속적으로 관리해야 할 데이터를 중심으로 속성을 정의한다.
- 속성의 가시성, 데이터 타입과 오퍼레이션의 가시성, 반환 타입은 향후 컴포넌트 식별 및 컴포넌트 인터페이스 식별에 중요한 기준이 되므로 정확하게 식별하며, 데이터 타입이나 반환 타입이 일반적으로 사용하는 기본 데이터 타입(Integer, String 등)이 아닌 경우 클래스 식별시 함께 식별해야 한다.
- 클래스간의 관계는 이후 컴포넌트 식별 및 컴포넌트 인터페이스의 상호작용 식별에서 활용할 수 있도록 정확한 관계 종류가 식별되어야 하며, 관계명(또는 역할명)과 관계의 다중성(multiplicity)이 식별되어야 한다.

■ 클래스 모델링 작업(1R34)

■ 클래스 모델링 결과와 이후 작업과의 관계

- 컨트롤, 엔티티 클래스는 컴포넌트의 내부 클래스로 유도되어 컴포넌트 식별(2D11)에서 컴포넌트 식별 및 아키텍처 설계의 기초가 된다. 또한 인터페이스 상호작용 정의(2D12)를 위한 내부 흐름을 도출한다.
- 엔티티 클래스는 데이터 모델링(2D15)에서 논리 엔티티로 유도될 수 있다.
- 바운더리 클래스는 사용자 인터페이스 명세(2D14)에서 화면(또는 보고서)의 정보 속성값과 오퍼레이션으로 유도된다.

■ 클래스 모델링시 표기법

- 클래스 모델링을 위한 표기법은 UML의 클래스 다이어그램 표기법을 사용하며, 필요시 패키지 다이어그램 표기법을 이용해 그룹핑한다.

■ 클래스 모델링 구성요소간 관계

- 클래스는 없거나 하나 이상의 속성과 오퍼레이션을 가지며, 오퍼레이션은 없거나 하나 이상의 파라미터를 가진다. 단, 속성, 오퍼레이션 또는 파라미터가 없는 것은 본 작업의 수행 수준에 따라 정의되지 않을 수 있음을 의미할 뿐이다.
- 클래스는 없거나 하나 이상의 객체를 가지며, 객체는 없거나 하나 이상의 객체간 관계를 가진다. 객체간 관계는 의미적으로 클래스의 오퍼레이션으로 추상화된다.
- 클래스는 없거나 하나 이상의 클래스간 관계를 가지며, 의미적으로 객체간 관계를 추상화 시킨 것이다.

클래스 모델링 작업(1R34)

[서식] 클래스 정의서(1R34a)

1. (클래스 다이어그램명)

1.1. 클래스 다이어그램

1.2. (클래스명)

1.2.1. 개요

1.2.2. 속성

속성명	가시성	타입	기본값	설명

1.2.3. 오퍼레이션

오퍼레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	

■ 클래스 모델링 작업(1R34)

■ [항목 설명] 클래스 정의서(1R34a)

1. (클래스 다이어그램 명)

1.1. 클래스 다이어그램

클래스와 클래스 간 관계를 클래스 다이어그램으로 작성한다.

1.2. (클래스명)

클래스 다이어그램에 포함된 클래스별로 작성한다. 하위 작성 항목 중 속성의 데이터 타입, 기본값과 오퍼레이션의 반환 타입, 파라미터 등 클래스 다이어그램에 포함해 작성한 항목은 중복 작성하지 않아도 무방하다.

1.2.1. 개요

클래스가 제공하는 기능과 처리 내역(오퍼레이션)에 초점을 두어 클래스의 개요를 작성한다.

1.2.2. 속성

해당 클래스의 속성별로 작성한다.

- 속성명
속성의 명칭을 기재한다.
- 가시성
해당 속성의 가시성을 기술한다.
- 타입
해당 속성의 데이터 타입을 기재한다.
- 기본값
객체 생성시 해당 속성의 기본값을 기재한다.
- 설명
속성에 대한 설명을 기술한다. 설명 작성시 '로그인 화면의 비밀번호', '부대부호를 임시로 저장하기 위한 속성' 등 속성이 의미하는 것에 초점을 두어 기술한다.

1.2.3. 오퍼레이션

해당 클래스의 오퍼레이션별로 작성한다.

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.

■ 클래스 모델링 작업(1R34)

- 가시성
해당 오퍼레이션의 가시성을 기재한다.
- 반환 타입
해당 오퍼레이션의 반환 데이터 타입을 기재한다.
- 파라미터 명칭
오퍼레이션 파라미터의 명칭과 데이터 타입을 기재한다.
- 파라미터 설명
오퍼레이션 파라미터의 설명을 기술한다. 파라미터 설명 작성시 해당 파라미터가 의미하는 것에 초점을 두어 기술한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 작성시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

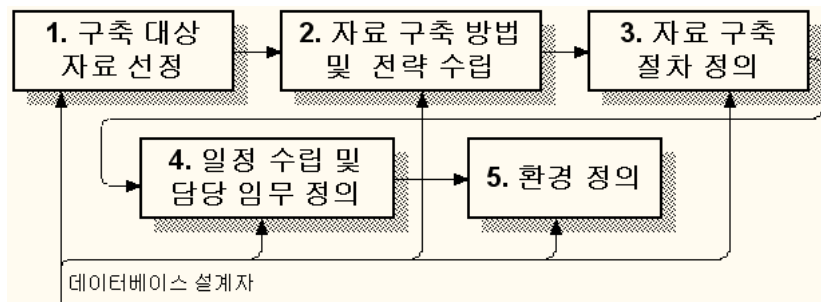
자료 구축 계획 작업(1R41)

개 요

요구사항을 바탕으로 시스템이 운용 이전에 구축되어야 할 데이터를 획득하기 위해 구축 대상 자료를 선정하고 자료의 획득 방법별 실제 구축을 위한 전략 등 세부 계획을 수립하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 현행 시스템 분석서 • 비즈니스 정의서 • 요구사항 정의서, (부록)요구사항 추적표
산출물	<ul style="list-style-type: none"> • 자료 구축 계획서
참여자	<ul style="list-style-type: none"> • 데이터베이스 설계자 <p>구축 대상 데이터의 특성에 따라 대상의 상태를 분석하고 전략, 절차, 일정 환경 등 세부 구축 계획을 수립한다.</p>
관련기법	해당 사항 없음.
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 구축 대상 자료 선정

자료 전환 및 구축 대상 자료를 선정한다.

- 구축 대상 자료 선정

현행 시스템 분석서, 비즈니스 정의서, 요구사항 정의서 등으로부터 시스템이 운용 이전에 구축되어야 할 데이터를 식별한다.

자료 구축 계획 작업(1R41)

- 원천 식별

식별된 구축 대상이 현재 어디에 존재하는지에 대한 탐색을 실시하고 실제 대상을 확인한다. 필요에 따라 현행 시스템 분석서나 타 시스템의 산출물 및 ERD를 참고하고 반드시 실제 시스템의 데이터베이스나 자료철을 육안으로 확인해야 한다.

2. 자료 구축 방법 및 전략 수립

식별된 구축 대상을 기반으로 구축 방법 및 전략을 수립한다.

- 자료 구축 방법 및 전략 수립

구축 대상 자료가 시스템의 데이터베이스로 존재해 전환이 가능한지, 신규로 구축이 필요한지 등을 파악하여 구축 방법을 식별하고, 해당 자료의 상태를 분석하여 적절한 부가 작업을 포함한 구축 전략을 수립한다. 경우에 따라 분산된 데이터의 통합을 위한 DI(Data Integration) 작업, 데이터 웨어하우징에 맞도록 변환하는 ETL(Extraction Transformation Loading) 과정, 정합성이 떨어지는 데이터의 Cleansing(정제) 작업을 포함해야 한다.

3. 자료 구축 절차 정의

자료 구축 대상과 방법에 맞도록 구축 작업 수행절차를 정의한다.

- 자료 구축 절차 정의

분석, 설계, 구현, 테스트의 단계에 맞도록 원천 자료의 상태를 분석하고 전환 또는 신규 구축하기 위한 데이터 스키마 및 자동화 프로그램의 설계, 전환 및 구축, 결과에 대한 검증 등의 자료 구축을 위한 세부 수행절차를 정의한다.

4. 일정 수립 및 담당 임무 정의

구축 대상별로 일정 및 담당 임무 정의 등의 구축 계획을 수립한다.

- 일정 수립 및 담당 임무 정의

자료 구축 절차에 맞는 일정과 각 절차를 담당하게 될 인원 및 조직을 결정하고 세부 임무를 정의한다.

■ 자료 구축 계획 작업(1R41)

5. 환경 정의

구축 대상별로 자료 구축을 위한 환경을 정의한다.

- 환경 정의

자료 구축을 위한 하드웨어 및 소프트웨어와 기타 제반 사항 등을 기술한다. 소프트웨어는 상용 소프트웨어뿐만 아니라 자체 설계 및 제작된 프로그램을 포함할 수 있다.

환경에 대한 정의는 자료 구축 작업뿐만 아니라 검증 또는 테스트 작업을 위한 환경까지 포함해서 기술한다.

■ 작업 수행시 고려사항

■ 자료 구축 계획 작업의 수행 여부

- 자료 구축 계획 작업은 구축 대상을 정의하고 분석 하는 작업으로써 개발 대상 시스템이 자료 구축 소요를 가질 경우에 한해 반드시 실시해야 한다.

■ 일정 수립의 유의점

- 일반적으로 자료 구축 일정은 프로젝트 성격에 따라 소프트웨어 개발 공정과 별도로 수립된다. 그러나 자료 구축은 소프트웨어 개발 공정의 데이터 모델링, 데이터베이스 설계 작업과 관련성이 있으므로 이를 고려하여 일정을 수립한다.

■ 기타 절차 및 산출물의 정의

- 데이터 통합 및 전환 작업은 사업 내에서 하나의 프로젝트로 구분될 수 있다. 따라서 이에 맞는 별도의 방법론 적용에 따라 절차와 산출물이 상이하다면 자료 구축 계획서의 절차 항목에 이를 기술하고 해당 산출물을 별첨 또는 별권으로 제출한다.

■ 자료 구축 계획 결과와 이후 작업과의 관계

- 자료 구축 계획 작업은 자료 구축 설계 작업(2D25)과 작업 시점을 고려하여 분리하였으나 관련성이 높으므로 2개 작업을 반복 수행하면서 자료 구축에 관한 분석 및 설계 결과의 완전성을 높여야 한다.

자료 구축 계획 작업(1R41)

[서식] 자료 구축 계획서(1R41a)

1. 자료 전환

1.5. (자료 전환 대상명)

1.1.1. 개요

1.1.2. 전환 대상 테이블

테이블 ID		테이블명		데이터 건수	
설 명					
자료 상태					
컬럼명	설명	키 유형	Not Null	데이터 타입 및 길이	제약조건

1.1.3. 전환 전략

1.1.4. 전환 절차

1.1.5. 전환 일정

1.1.6. 담당 조직 및 세부 임무

1.1.7. 전환 환경

2. 초기 자료 구축

2.1. (자료 구축 대상명)

2.1.1. 개요

2.1.2. 구축 전략

자료 구축 계획 작업(1R41)

2.1.3. 구축 방법

자료명	자료 건수	원천 형태	구축 방법

2.1.4. 구축 절차

2.1.5. 구축 일정

2.1.6. 담당 조직 및 세부 임무

2.1.7. 구축 환경

자료 구축 계획 작업(1R41)

■ [항목 설명] 자료 구축 계획서(1R41a)

1. 자료 전환

자료의 획득 방법 중 자료 전환은 기존 시스템 또는 타 시스템에서 기 구축된 데이터베이스 데이터를 개발 대상 시스템의 데이터베이스에 맞게 전환 및 이관하는 것이다.

1.1. (자료 전환 대상명)

자료 전환 대상명은 자료를 비즈니스 관점으로 묶어 기술한다. 하나의 대상은 여러 테이블로 구성될 수 있다.

1.1.1. 개요

전환 대상 자료의 소유권, 운용 방식, 탑재 시스템, DBMS 명칭, 분산 방식, 비즈니스 기능 및 목적, 자료 상태 등을 기술한다.

1.1.2. 전환 대상 테이블

전환 대상인 타 시스템의 데이터 테이블에 대한 명세를 작성한다.

- 테이블ID
테이블을 식별할 수 있는 식별자를 기재한다.
- 테이블명
테이블을 대표하는 명칭을 기재한다.
- 데이터 건수
전환 대상의 개략적 데이터 건수를 기재한다.
- 설명
테이블에 대한 비즈니스 활용성에 대한 간단한 설명을 기술한다.
- 자료 상태
정합성, 운용됐던 기간 등 데이터의 자료로서의 상태를 기술한다.
- 컬럼명
컬럼의 명칭을 기재한다.
- 설명
컬럼에 대한 간단한 설명 또는 논리 엔티티 속성명을 기재한다.
- 키 유형
컬럼의 키 여부와 유형으로써 주키(PK : Primary Key), 외래키(FK : Foreign Key)를 기재하고 주키일 경우 주키 순서 번호를 함께 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.

자료 구축 계획 작업(1R41)

- 데이터 타입 및 길이
DBMS에 맞는 컬럼의 데이터 타입 및 길이를 기재한다.
- 제약조건
필요 시 각 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.

1.1.3. 전환 전략

전환 대상 자료의 개요를 고려하여 이에 맞는 전환 전략을 수립하여 기술한다.

1.1.4. 전환 절차

전환 전략에 부합하는 세부 절차를 기술한다.

1.1.5. 전환 일정

전환을 위한 일정을 단계로 구분하여 기술한다. 전환 전략 및 절차와 함께 기술할 수 있다.

1.1.6. 담당 조직 및 세부 임무

전환을 담당할 조직과 담당자를 기재하고 각 조직 및 담당자의 세부 임무를 기술한다. 전환 전략 및 절차와 함께 기술할 수 있다.

1.1.7. 전환 환경

전환에 필요한 물리적 요소(장소, 장비, 소프트웨어 등)를 기술하고 필요시 구성도를 작성한다.

2. 초기 자료 구축

자료의 획득 방법 중 초기 자료 구축은 아직 전산 데이터베이스로 구축되어 시스템에서 활용되지 못하는 자료(단순 전산화된 데이터 포함)와 신규로 구축 필요성이 식별된 자료를 개발 대상 시스템의 데이터베이스에 맞게 변환 또는 새로 입력하는 것이다.

2.1. (자료 구축 대상명)

자료 구축 대상명은 자료를 비즈니스 관점으로 묶어 작성한다. 하나의 대상은 여러 테이블로 구성될 수 있다.

자료 구축 계획 작업(1R41)

2.1.1. 개요

구축 대상 자료의 성격(제원성, 운용성 등), 비즈니스 기능 및 목적 등을 기술한다.

2.1.2. 구축 전략

구축 대상 자료의 특성을 고려하여 이에 맞는 구축 전략을 기술한다.

2.1.3. 구축 방법

- 자료명
구축 자료를 구성하는 하나의 집합성 자료(향후 하나의 테이블로 구성될 수 있는 후보)를 대표하는 명칭을 기재한다.
- 자료 건수
하나의 집합성 자료에 대해 초기에 구축할 개략적 건수를 기재한다.
- 원천 형태
자료의 원천이 어떠한 형태로 존재하는지 기재한다. (예 : 책자, 스프레드시트, 워드 프로세서 등)
- 구축 방법
원천 자료를 전산 데이터 형태로 변환하여 구축할 방법에 대해서 기술한다. (예 : 입력, 스캔 및 보정, 촬영, 변환 기능 이용, 변환 프로그램 개발, 수기 입력 등)

2.1.4. 구축 절차

구축 전략에 부합하는 세부 절차를 기술한다. 구축 대상량이 많을 경우 분할 작업 및 통합/검증 방법 등이 반영되도록 한다.

2.1.5. 구축 일정

구축을 위한 일정을 단계로 구분하여 기술한다. 구축 절차와 함께 기술할 수 있다.

2.1.6. 담당 조직 및 세부 임무

구축을 담당할 조직과 담당자를 기재하고 각 조직 및 담당자의 세부 임무를 기술한다. 구축 절차와 함께 기술할 수 있다.

2.1.7. 구축 환경

구축에 필요한 물리적 요소(장소, 장비, 소프트웨어 등)를 기술하고 필요시 구성도를 작성한다.

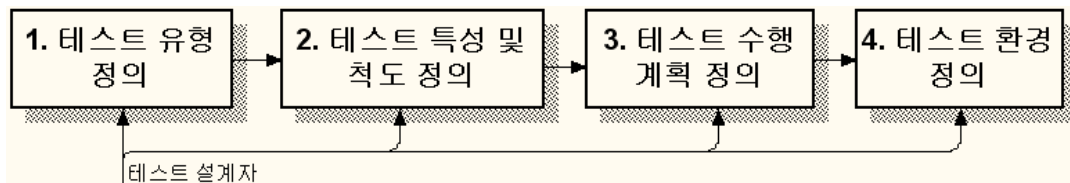
테스트 계획 작업(1R51)

개요

개발 대상 시스템의 결함을 찾아내고 요구사항을 충족하는지를 판단하는 테스트를 수행하기 위해 일정, 수행내용 및 테스트 환경 등 전반적인 계획을 수립하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 요구사항 정의서, (부록)요구사항 추적표 유스케이스 정의서
산출물	<ul style="list-style-type: none"> 테스트 계획서
참여자	<ul style="list-style-type: none"> 테스트 설계자 <p>개발 대상 시스템의 특성에 따라 테스트 유형과 일정 등의 세부 계획을 수립한다.</p>
관련기법	<ul style="list-style-type: none"> 경험 기반 테스트 구조 기반 테스트 명세 기반 테스트

절차 흐름도



절차 설명

1. 테스트 유형 정의

수행할 테스트의 종류와 각 종류별 유형을 정의한다. 본 방법론의 테스트 종류는 단위 테스트, 소프트웨어 통합 테스트, 시스템 통합 테스트로 구분되며, 프로젝트 특성에 따라 별도의 테스트 활동을 추가할 수 있다.

• 테스트 유형 정의

개발 대상 시스템의 크기 및 범위를 고려해 시스템의 완전성을 판단할 수 있도록 테스트의 유형을 분류하고 각 유형에 대한 정의를 수행한다. 테스트 유형이 정의되면 해당 테스트에 대한 테스트케이스를 어떠한 기준으로 도출할 것인지를 정의한다.

■ 테스트 계획 작업(1R51)

2. 테스트 특성 및 척도 정의

테스트케이스 도출과 별개로 설계 및 구현 결과를 기준으로 하여 실제 수행한 테스트의 충분 정도를 판단하고 평가하기 위한 척도를 정의한다.

- 테스트 특성 및 척도 정의

테스트 유형별로 테스트 특성을 정의하고 각 테스트 특성을 측정할 수 있는 기준과 통과 기준을 정의한다.

3. 테스트 수행 계획 정의

개략적인 테스트 일정 및 담당 임무를 정의해 테스트 계획을 수립한다.

- 테스트 수행 계획 정의

각 테스트 유형별로 테스트를 수행할 일정과 담당자, 테스트 환경을 정의하되 대상별로 분리하여 수행하는 것이 필요하면 이를 구분하여 정의한다.

4. 테스트 환경 정의

테스트별로 주요 테스트 환경을 정의한다.

- 테스트 환경 정의

각 테스트 종류별로 테스트를 수행할 환경으로써 소프트웨어와 하드웨어, 네트워크 등을 기술한다.

■ 작업 수행시 고려사항

■ 테스트 계획 작업의 수행 여부

- 테스트 계획 작업은 테스트 대상을 정의하고 분석 하는 작업으로써 테스트 준비 및 수행 작업에 영향을 미치나, 소규모 사업의 경우 각 테스트 준비 작업과 병행하여 실시가 가능하므로 생략이 가능하다.

■ 테스트 유형의 정의

- 테스트 유형은 구현 결과의 어떠한 능력 또는 특성을 시험함으로써 결함을 찾아낼 것인지에 대한 정의로써 시스템의 범위, 개발 내용 및 구현 방법의 특성에 따라 정의되어야 한다.

테스트 계획 작업(1R51)

- 어떠한 테스트든지 시스템의 전부를 커버할 수 없다는 것을 유의해야 한다. 예를 들어 간단한 코드의 경우는 모든 코드와 분기(branch)에 대해 테스트케이스를 추출하는 것이 가능하다. 그러나 매우 복잡한 경우는 모든 경우에 따른 테스트케이스를 추출하는 것과 그대로 수행하는 것이 불가능하며, 하나의 테스트케이스에 대한 테스트를 종료하지 않고 무한히 수행하는 것도 불가능하다. 따라서 테스트 일정과 자원을 고려해서 평가할 테스트 유형을 선정하고 측정 기준과 통과 기준을 설정해야 한다.

■ 단위 테스트 유형 정의의 예

- 단위 테스트는 소프트웨어의 기능 수행 단위를 정하여 해당 범위에 대해 결합 여부를 파악하기 위해 테스트를 수행하는 것이다. 크게 사용자 인터페이스에 대한 테스트와 컴포넌트에 대한 테스트, 데이터베이스 프로그램에 대한 테스트 등으로 구분될 수 있다.
 - 사용자 인터페이스 테스트는 사용자와 시스템간의 인터페이스 층을 담당하면서 데이터의 기본적인 유효성을 체크해줄 수 있는가(유효성 테스트), 사용자 인터페이스가 가지는 오퍼레이션이 정확하게 동작하는가(오퍼레이션 테스트), 사용자에게 친숙한 안내와 조작 방법을 보유하고 있는가(사용성 테스트)에 대한 테스트를 할 수 있다.
 - 컴포넌트 테스트는 단위 메소드가 정확하게 동작하는가(단위 메소드 테스트), 메소드가 다른 메소드와 정확하게 연결되는가(메소드 상호관계 테스트), 클래스가 다른 클래스를 정확히 참조하거나 상속하는가(클래스 상호관계 테스트), 컴포넌트가 입력값에 대해 정확히 처리하여 출력값을 리턴하는가(컴포넌트 인터페이스 테스트)에 대한 테스트를 할 수 있다.
 - 데이터베이스 프로그램에 대한 테스트는 함수, 프로시저 등 DBMS에 삽입되어 동작하는 프로그램이 정확하게 동작하는가(동작 테스트)에 대한 테스트를 할 수 있다.

■ 소프트웨어 통합 테스트 유형 정의의 예

- 소프트웨어 통합 테스트는 단위 테스트가 완료된 컴포넌트와 비 컴포넌트(예 : 사용자 인터페이스 파일, 배치 파일 등) 등 소프트웨어의 구성 항목을 통합하고 통합 이상 여부와 결합을 파악하기 위해 테스트를 수행하는 것이다.
- 소프트웨어 통합 테스트에서는 통합된 컴포넌트/비 컴포넌트 묶음이 입력값에 대해 정확히 처리하여 출력값을 리턴하는가(컴포넌트 인터페이스 테스트), 컴포넌트의 내부 로직이 외부 변화 없이 변경되었을 경우에 연결된 컴포넌트에서 오류가 발생하는가(컴포넌트 의존성 테스트), 컴포넌트의 외부 환경이 인

테스트 계획 작업(1R51)

터페이스의 변화 없이 변경되었을 경우 컴포넌트에서 오류가 발생하는가(서브 시스템 의존성 테스트)에 대한 테스트를 할 수 있다.

■ 시스템 통합 테스트 유형 정의의 예

- 시스템 통합 테스트는 소프트웨어 통합 테스트가 완료된 소프트웨어와 하드웨어, 데이터, 네트워크 등을 통합하고 통합 이상 여부와 결함의 유무, 요구사항의 충족 여부를 파악하기 위해 테스트를 수행하는 것이다.
- 또한 시스템 통합 테스트에서는 시스템이 기능적 요구사항을 만족하는가, 보안이나 성능 등의 비기능적 요구사항을 만족하는가에 대한 테스트를 할 수 있다.

■ 테스트 계획 결과와 이후 작업과의 관계

- 테스트 종류 및 유형은 각 테스트 준비 작업의 테스트케이스로 유도된다.
- 테스트 특성 및 척도는 각 테스트 수행 작업의 평가로 유도된다.

■ 테스트 계획 작업(1R51)

■ [서식] 테스트 계획서(1R51a)

1. 테스트 종류 및 유형

테스트 종류	테스트 유형	수행 내용	테스트케이스 선정 기준
단위 테스트			
소프트웨어 통합 테스트			
시스템 통합 테스트			

2. 테스트 특성 및 척도

테스트 종류	테스트 유형	테스트 유형 특성			
		특성명	특성 설명	측정 기준	통과 기준
단위 테스트					

테스트 계획 작업(1R51)

테스트 종류	테스트 유형	테스트 유형 특성			
		특성명	특성 설명	측정 기준	통과 기준
소프트웨어 통합 테스트					
시스템 통합 테스트					

3. 테스트 수행 계획

테스트 종류	테스트 유형	대상	일정	담당자	테스트 환경
단위 테스트					

■ 테스트 계획 작업(1R51)

테스트 종류	테스트 유형	대상	일정	담당자	테스트 환경
소프트웨어 통합 테스트					
시스템 통합 테스트					

4. 테스트 환경

■ 테스트 계획 작업(1R51)

■ [항목 설명] 테스트 계획서(1R51a)

1. 테스트 종류 및 유형

- 테스트 종류
단위 테스트, 소프트웨어 통합 테스트, 시스템 통합 테스트 중 해당되는 테스트를 기술한다. 또한, 프로젝트 특성에 따라 추가된 테스트 활동이 있으면 해당되는 테스트 명을 기술한다.
- 테스트 유형
각 시스템 특성에 따라 테스트 유형을 정의한다.
예를 들어 단위 테스트에서는 컴포넌트의 경우 단위 메소드 테스트, 메소드 상호관계 테스트, 클래스 상호관계 테스트, 컴포넌트 인터페이스 테스트 등이 될 수 있으며 사용자 인터페이스의 경우 유효성 테스트, 오퍼레이션 테스트, 사용성 테스트 등이 될 수 있고, DB 프로그램의 동작 정확성 테스트 등이 될 수 있다.
소프트웨어 통합 테스트에서는 컴포넌트 인터페이스 테스트, 컴포넌트 의존성 테스트, 서브시스템 의존성 테스트 등이 될 수 있다.
시스템 통합 테스트에서는 시스템 기능 테스트, 시스템 비기능 테스트 등이 될 수 있다.
- 수행내용
해당 테스트의 세부적인 범위와 주요 수행 내용에 대해 기술한다.
- 테스트케이스 선정 기준
테스트 유형 특성별로 테스트케이스를 선정하기 위한 기준을 기술한다.

2. 테스트 특성 및 척도

- 테스트 종류
테스트 종류 및 유형에서 정의한 테스트 종류를 기술한다.
- 테스트 유형
테스트 종류 및 유형에서 정의한 테스트 유형을 기술한다.
- 특성명
테스트 결과를 평가하기 위해 테스트 유형이 어떤 특성으로 정의될 수 있는지를 기재한다.
- 특성 설명
특성명에 대한 개략적인 설명을 기술한다.
- 측정기준
테스트 유형 특성에 대한 측정기준을 기재한다.
- 통과기준
측정기준에 따른 테스트의 통과 기준을 기재한다.

테스트 계획 작업(1R51)

3. 테스트 수행 계획

- 테스트 종류
테스트 종류 및 유형에서 정의한 테스트 종류를 기재한다.
- 테스트 유형
테스트 종류 및 유형에서 정의한 테스트 유형을 기재한다.
- 대상
테스트를 구분하여 수행할 경우 관련 대상 명칭을 기재한다.
- 일정
각 테스트 별로 수행 기간을 기재한다. 컴포넌트 테스트의 경우 구현과 동시에 수행할 경우 일정을 생략할 수 있다.
- 담당자
각 테스트에 대한 담당자명을 기재한다.
- 테스트 환경
각 테스트가 수행될 환경을 식별할 수 있는 명칭을 기재한다.

4. 테스트 환경

각 테스트 환경을 구성하는 하드웨어와 소프트웨어, 네트워크 등을 기술한다.

설계 단계(2D)

개요

설계 단계는 분석 결과를 바탕으로 요구사항을 어떻게 시스템으로 구현할 것인지를 설계하고 구현된 시스템을 설치하기 위한 계획을 수립하는 것을 목적으로 한다. 설계 단계는 개략 설계와 상세 설계, 시스템 설치 준비 활동으로 구성된다. 구현 플랫폼을 전제로 시스템의 소요가 제기되었거나 구현 플랫폼에 종속적인 기능이 존재하는 등 설계 이전에 구현 플랫폼이 결정되었을 경우 방법론 테일러링을 통해 개략 설계와 상세 설계 활동을 병행 또는 통합해 수행한다.

활동/작업 설명

1. 개략 설계(2D1)

개략 설계 활동은 컴포넌트의 식별 및 명세를 수행하고 사용자 인터페이스의 제어 구조와 흐름을 설계하며, 논리 데이터 모델을 설계한다. 개략 설계 활동은 구현 플랫폼에 대한 고려를 가능한 한 지양하고 시스템의 기능을 어떤 로직을 이용해 제공할 것인가에 초점을 둔다.

작업ID	작업명	작업내용
2D11	컴포넌트 식별	요구사항 분석 결과를 기반으로 컴포넌트를 식별하고 컴포넌트 아키텍처를 설계한다.
2D12	인터페이스 상호작용 명세	유스케이스 모델에서 분석된 액터와 시스템간의 상호작용을 기반으로 컴포넌트가 제공하는 서비스를 명확히 하고 컴포넌트간의 호출관계를 정제한다.
2D13	컴포넌트 명세	컴포넌트가 제공하는 기능과 인터페이스를 명세하고 컴포넌트 내부의 처리 로직을 설계한다.
2D14	사용자 인터페이스 명세	사용자 인터페이스의 구성요소를 식별해 제어 구조를 모델링하고 사용자 인터페이스 구현을 위한 구성요소를 설계한다.
2D15	데이터 모델링	시스템이 관리해야 할 데이터를 논리 데이터 모델로 작성한다.

설계 단계(2D)

2. 상세 설계(2D2)

상세 설계 활동은 컴포넌트 및 사용자 인터페이스를 구현 플랫폼에 매핑하고 논리 데이터 모델을 DBMS에 따라 데이터베이스화 한다. 또한 구현 플랫폼 매핑에 따라 추가되는 기능을 설계하고 시스템 운용에 필요한 사용자 시스템간 인터페이스, 자료 구축 및 시스템 보안 설계를 수행한다.

작업ID	작업명	작업내용
2D21	컴포넌트 구현 설계	개략 설계에서 명세한 컴포넌트를 구현 플랫폼에 따라 매핑하고 추가 기능을 설계한다. 또한, 컴포넌트가 시스템 운용 환경에 물리적으로 배치되는 모습을 설계한다.
2D22	사용자 인터페이스 구현 설계	개략 설계에서 명세한 사용자 인터페이스 구성요소 및 제어 구조를 구현 플랫폼에 따라 매핑하고 시스템 운용 환경에서 물리적으로 배치되는 모습을 설계한다.
2D23	데이터베이스 설계	DBMS가 제공하는 메커니즘에 따라 논리 데이터 모델을 물리 데이터 모델로 변환하고 DB 및 DBMS 내장 프로그램을 설계한다.
2D24	연동 설계	시스템간 인터페이스를 구현하기 위한 기능을 설계한다.
2D25	자료 구축 설계	초기 운용 자료 및 시스템 운용에 필요한 기준 자료를 데이터베이스 설계 결과에 따라 매핑하고 자료 구축을 위한 소프트웨어를 설계한다.
2D26	시스템 보안 설계	시스템에서 관리해야 할 보안 대상을 식별하고 보안 영역과 요소별로 보안 대상을 배치해 시스템 운용시 적용할 시스템 보안 내역을 설계한다.

설계 단계(2D)

3. 시스템 설치 준비(2D3)

시스템 설치 준비 활동은 개발된 시스템을 설치하기 위한 계획을 수립한다.

작업ID	작업명	작업내용
2D31	시스템 설치 계획	개발된 시스템을 설치하기 위한 일정, 자원 및 제반 절차를 계획한다.

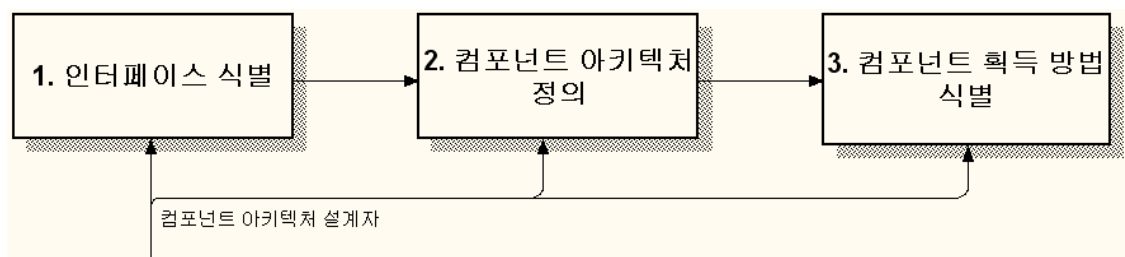
컴포넌트 식별 작업(2D11)

개요

유스케이스 모델링 및 클래스 모델링 결과를 기반으로 다양한 방식의 컴포넌트 식별 기준을 적용해 컴포넌트를 식별하고 컴포넌트 아키텍처를 설계하는 작업이다. 또한, 식별한 컴포넌트에 대해 재사용 컴포넌트(기존(legacy), 상용(COTS), 관급(GOTS) 등) 활용 가능성을 판단해 컴포넌트 획득 방법을 정의하고 재사용 가능한 컴포넌트가 있을 경우 활용 및 연계 방법을 정의하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 유스케이스 정의서 클래스 정의서
산출물	<ul style="list-style-type: none"> 컴포넌트 아키텍처 명세서
참여자	<ul style="list-style-type: none"> 컴포넌트 아키텍처 설계자 컴포넌트를 식별하고 컴포넌트 아키텍처를 설계한다.
관련기법	<ul style="list-style-type: none"> UML 핵심객체 중심 인터페이스 식별 UDA 클러스터링 디자인 패턴
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 공통운용환경(COE)

절차 흐름도



절차 설명

1. 인터페이스 식별

컴포넌트 인터페이스를 식별하고 이를 기반으로 컴포넌트를 도출한다. 인터페이스는 컴포넌트가 외부에 제공하는 서비스의 접근 경로이다. 컴포넌트 외부에서

컴포넌트 식별 작업(2D11)

는 인터페이스를 통해서만 컴포넌트의 서비스를 제공받을 수 있으므로 인터페이스의 식별은 컴포넌트의 범위를 결정하는 기준이 된다.

- 인터페이스 식별
소프트웨어 아키텍처, 유스케이스 모델링 및 클래스 모델링 결과를 기반으로 각 결과물의 구성 요소 간 상호작용 내용을 식별하고 이를 기반으로 컴포넌트의 인터페이스를 식별한다.
- 컴포넌트 식별
식별된 인터페이스별로 컴포넌트가 제공해야 하는 서비스를 도출해 컴포넌트를 식별하고 인터페이스가 제공해야 하는 오퍼레이션을 정의한다.

2. 컴포넌트 아키텍처 정의

컴포넌트간의 상호작용 설계 결과와 소프트웨어 아키텍처를 기반으로 컴포넌트 인터페이스 및 컴포넌트를 정제하고 컴포넌트간의 호출 횟수 최소화, 양방향 의존성 제거, 아키텍처 패턴의 적용 등을 고려하여 컴포넌트 아키텍처를 정의한다.

- 컴포넌트 상호작용 식별
시퀀스(또는 커뮤니케이션) 다이어그램을 이용해 인터페이스 오퍼레이션을 중심으로 유스케이스 사건흐름(또는 시나리오)을 모델링해 컴포넌트간의 상호작용을 개략적으로 설계한다. 또한 상호작용 설계를 통해 인터페이스의 오퍼레이션별로 제약사항이나 전제조건을 식별한다.
- 컴포넌트 아키텍처 정의
컴포넌트간의 상호작용 및 소프트웨어 아키텍처를 기반으로 컴포넌트를 배치해 컴포넌트 아키텍처를 정의한다.

3. 컴포넌트 획득 방법 식별

식별된 컴포넌트를 기반으로 재사용 가능한 컴포넌트를 확인한다.

- 재사용 컴포넌트 확인
기존(legacy), 상용(COTS), 관급(GOTS) 등 재사용 가능한 컴포넌트 정보를 수집하고 각 재사용 컴포넌트 명세를 식별된 컴포넌트와 비교해 재사용 여부를 결정한다.
- 재사용 컴포넌트 연계 방법 정의
재사용 컴포넌트를 식별하면서 재사용 컴포넌트 사용을 위한 아답터(adapter), 랩퍼(wrapper) 클래스 등 연계 방법을 정의한다.

컴포넌트 식별 작업(2D11)

작업 수행시 고려사항

■ 컴포넌트 식별의 의미

- 컴포넌트 식별 작업은 시스템 외부와 상호작용하는 UI 컴포넌트, 시스템 내부 컴포넌트간의 상호작용하는 비즈니스 컴포넌트를 식별하고 컴포넌트 아키텍처를 설계하는 것을 목적으로 한다.
- 컴포넌트 식별 작업의 결과물인 컴포넌트 식별 결과와 컴포넌트 아키텍처는 완전하지 않으며, 개략 설계(2D1) 활동 내에서 지속적으로 정제해야 한다.
- 컴포넌트 식별 작업에서는 컴포넌트의 내부 로직 등 세부적인 사항은 고려하지 않으며, 상세 사항은 이후 작업을 통해 설계된다. 그러나 컴포넌트간의 상호작용 관계와 시스템 외부와의 상호작용 관계를 명확히 하기 위해 인터페이스가 제공하는 기본적인 오퍼레이션과 해당 오퍼레이션이 제공하는 서비스는 정의되어야 한다.

■ 인터페이스 식별시 고려사항

- 다수의 컴포넌트 기반 방법론에서 인터페이스 식별을 통해 컴포넌트의 경계를 정의하고 이를 바탕으로 컴포넌트를 식별한다.
- 컴포넌트 인터페이스를 식별하는 방법은, 데이터 관점에서 핵심 클래스를 식별하고 이를 중심으로 관련 객체를 묶는 방법, 공통된 유스케이스로 식별하는 방법, 유스케이스와 엔티티 클래스와의 상관도를 분석하는 UDA(Usecase Data Access) 클러스터링 방법, 가변성 여부를 고려해 가변성이 요구되는 비즈니스 룰이나 알고리즘 관점으로 식별하는 방법 등 다양한 방법이 활용되고 있다.
- 인터페이스 식별 방법별로 수행 절차의 차이는 있으나 다양한 방법을 혼용해 인터페이스를 식별한다는 전제하에 다음의 항목을 수행한다.

수행 내용	설 명
유스케이스 관점에서 비즈니스 컴포넌트의 주요 접근 경로를 식별	시스템 외부와의 상호작용은 UI 컴포넌트의 인터페이스로 식별될 수 있으며, 소프트웨어 아키텍처에 따라 별도의 UI 컴포넌트를 식별하지 않는 경우에는 비즈니스 컴포넌트의 인터페이스를 식별하는 기준이 된다.
	유스케이스 모델링 결과를 기반으로 하나의 액터가 상호작용하는 유스케이스를 묶어 컴포넌트로 식별하거나 액터들이 공통적으로 상호작용하는 유스케이스를 묶어 컴포넌트를 식별할 수 있다.
	유스케이스간의 관계를 기반으로 공통 유스케이스를 별도의 컴포넌트로 식별하거나 확장 유스케이스를 묶어 컴포넌트로 식별한다.

컴포넌트 식별 작업(2D11)

수행 내용	설 명
유스케이스의 사건 흐름과 가변성을 기반으로 비즈니스 규칙을 식별	비즈니스 규칙은 컴포넌트에서 처리되어 인터페이스를 통해 다른 컴포넌트 또는 시스템 외부에 제공되므로 비즈니스 규칙이 적용되는 대상은 컴포넌트로, 비즈니스 규칙별 가변성은 인터페이스로 식별한다.
엔티티 클래스의 상호작용을 식별	컴포넌트의 대체성과 재사용성을 높이기 위해서는 향후 DB화되는 엔티티 클래스를 기반으로 데이터의 생성, 삭제, 수정, 조회 여부를 판단해 엔티티 클래스로의 상호작용을 인터페이스로 식별하고 응집도가 높은 클래스를 그룹핑하여 컴포넌트를 식별한다.
엔티티 클래스를 기반으로 핵심 클래스를 선정	엔티티 클래스 중 해당 비즈니스 모델에서 중요하고 핵심적인 역할과 책임을 갖는 핵심(Core Type) 클래스를 식별하고 이 핵심 클래스를 중심으로 응집도가 높은 클래스를 그룹핑해 인터페이스를 식별하고 컴포넌트를 도출한다.

- 컴포넌트의 인터페이스 식별 방법이 다양한 이유는 적절한 컴포넌트의 식별 과정이 쉽지 않기 때문이다. 따라서 유일한 최선의 방법이 존재하지 않으므로 다양한 기법을 혼용하여 컴포넌트를 식별해야 한다.
- 컴포넌트는 대체성과 재사용성이 높고 응집도가 높은 독립적인 배포 가능한 단위이다. 따라서 컴포넌트 식별시 컴포넌트에서 활용되는 데이터를 고려해 컴포넌트의 데이터 접근 경로를 단일화/단순화하거나 데이터를 중심으로 컴포넌트를 식별해야 한다.
- 하나의 컴포넌트에 반드시 하나의 인터페이스가 존재하는 것은 아니며, 컴포넌트의 가변성, 상호작용하는 대상의 수에 따라 하나의 컴포넌트에 여러 개의 인터페이스가 존재할 수 있다.
- 인터페이스 및 컴포넌트 식별시 다음의 사항을 고려해 수행한다.
 - 컴포넌트는 대체성이 높고 재사용이 용이한가?
 - 컴포넌트는 다른 컴포넌트에 최소한의 의존성만을 갖고 있는가?
 - 컴포넌트는 다른 컴포넌트와 역할 구분이 명확한가?
 - 컴포넌트가 요구되는 도메인이나 데이터에 대한 응집성이 높은가?
 - 하나의 컴포넌트가 너무 많은 서비스를 제공하지는 않는가?
 - 컴포넌트 내부의 변경으로 인한 외부의 변경이 최소화 되었는가?
 - 컴포넌트가 제공하는 서비스가 인터페이스를 통해 모두 제공되는가?
 - 인터페이스가 제공하는 기능이 명확히 표현되었는가?
- 컴포넌트간 상호작용은 시스템 외부의 액터가 시스템을 통해 수행하고자 하는 일련의 절차로써 컴포넌트간의 상호작용은 가능한 한 양방향 참조, 순환 참조, 네트워크 참조는 지양하고 단방향으로만 관계를 설정하도록 한다.

컴포넌트 식별 작업(2D11)

■ 컴포넌트 아키텍처 정의시 고려사항

- 컴포넌트 아키텍처는 소프트웨어 아키텍처를 기반으로 정의되어야 하며, 컴포넌트 아키텍처가 정의된 이후에는 가능한 컴포넌트 아키텍처의 전체 구조를 수정하지 않도록 주의해야 한다.
- 컴포넌트 아키텍처의 전체 구조에 변경이 필요할 경우 시스템 아키텍처 정의(1R21) 작업부터의 재수행을 고려해야 한다.

■ 컴포넌트 획득 방법 식별시 고려사항

- 재사용 컴포넌트는 그래프 작성, 인증, 수식 처리 등의 단일 기능을 제공하는 것부터 특정 기능군 또는 도메인 전체를 아우르는 등 배포 단위나 크기가 다양하다. 또한, 시스템의 제약 사항이나 비기능적 요구사항과 차이가 있어 소프트웨어 아키텍처 전체를 변경해야 할 경우도 있으므로 사용 결정시 주의해야 한다.
- 재사용할 컴포넌트 선정 시 ‘국방상호운용성관리 지시’에 따라 국방 공통컴포넌트의 활용을 우선적으로 고려해야 한다. 이때 국방 공통컴포넌트의 기능 충족도와 컴포넌트 아키텍처 상에 문제가 없는지, 기술적 제약 사항은 없는지 확인해야 한다.
- 컴포넌트 식별 및 컴포넌트 아키텍처 정의는 재사용 컴포넌트를 고려하지 않고 수행하는 절차이므로 만일 기존의 컴포넌트를 재사용하여 선행 절차의 결과물의 변경이 수반될 경우 컴포넌트 식별 결과와 컴포넌트 아키텍처 정의부터 절차를 재수행한다.

■ 컴포넌트 식별 결과와 이후 작업과의 관계

- 컴포넌트 식별(2D11)과 인터페이스 상호작용 정의(2D12), 컴포넌트 명세(2D13)는 투입 노력과 범위를 고려해 작업이 구분되었으나 작업간의 관련성이 높으므로 3개 작업을 반복 수행하면서 컴포넌트 식별 및 명세의 완전성을 높여야 한다.
- 컴포넌트 식별 완료 후 그 결과를 요구사항 추적표에 반영하여 유스케이스와 컴포넌트간의 관계 확인이 가능하도록 한다.

■ 컴포넌트 식별시 표기법

- 컴포넌트 아키텍처를 위한 표기법은 UML의 컴포넌트 다이어그램 표기법을 사용한다. 이 외에도 일반적인 구성도의 모습도 가능하며, 컴포넌트 간 참조 또는 의존 관계를 표현할 수 있는 표기법을 사용하도록 한다.

■ 컴포넌트 식별 작업(2D11)

■ 컴포넌트 식별 구성요소간 관계

- 컴포넌트는 하나 이상의 인터페이스를 포함하며, 인터페이스는 하나의 컴포넌트에 포함된다.
- 컴포넌트 간 관계는 컴포넌트와 컴포넌트 인터페이스간의 관계이며, 인터페이스의 목적상 인터페이스는 반드시 하나 이상의 컴포넌트와 관계가 있다.
- 인터페이스는 하나 이상의 오퍼레이션을 포함한다.

컴포넌트 식별 작업(2D11)

[서식] 컴포넌트 아키텍처 명세서(2D11a)

1. 컴포넌트 목록

식별자	컴포넌트명	설명	인터페이스		
			명칭	오퍼레이션 명	오퍼레이션 설명

2. 컴포넌트 아키텍처 다이어그램

3. 재사용 컴포넌트 활용 방법

식별자	컴포넌트명	획득 방법	연계 방법	부가사항

컴포넌트 식별 작업(2D11)

■ [항목 설명] 컴포넌트 아키텍처 명세서(2D11a)

1. 컴포넌트 목록

식별한 컴포넌트 목록을 작성한다. 컴포넌트 아키텍처 다이어그램에 컴포넌트 목록의 항목을 포함해 작성하였다면 작성하지 않아도 무방하다.

- 식별자
해당 컴포넌트의 식별자를 기재한다. 컴포넌트명을 식별자로 사용할 경우 생략해도 무방하다.
- 컴포넌트명
해당 컴포넌트의 명칭을 기재한다.
- 설명
해당 컴포넌트가 제공하는 주요 서비스, 제어하는 데이터, 수행을 위해 참조하는 데이터를 중심으로 설명을 기술한다. 재사용 컴포넌트에 포함된 별도의 명세가 있을 경우 해당 컴포넌트 명세로 대체할 수 있다.
- 인터페이스 명칭
해당 컴포넌트가 제공하는 인터페이스 명칭을 기재한다.
- 인터페이스 오퍼레이션명
해당 인터페이스의 오퍼레이션 명칭을 기재한다. 재사용 컴포넌트의 경우 활용할 오퍼레이션을 모두 기술한다. 재사용 컴포넌트에 포함된 별도의 명세가 있을 경우 해당 컴포넌트 명세로 대체할 수 있다.
- 인터페이스 오퍼레이션 설명
해당 오퍼레이션을 통해 제공받을 수 있는 서비스가 무엇인지에 초점을 두어 기술한다. 재사용 컴포넌트에 포함된 별도의 명세가 있을 경우 해당 컴포넌트 명세로 대체할 수 있다.

2. 컴포넌트 아키텍처 다이어그램

컴포넌트 아키텍처 다이어그램을 작성한다. 컴포넌트 아키텍처 다이어그램에 대한 부가 설명이 필요할 경우 다이어그램 하단에 작성한다.

3. 재사용 컴포넌트 활용 방법

모든 재사용 컴포넌트에 대해 활용 방법을 작성한다.

- 식별자
해당 컴포넌트의 식별자를 기재한다. 컴포넌트명을 식별자로 사용할 경우 생략해도 무방하다.

■ 컴포넌트 식별 작업(2D11)

- 컴포넌트명
해당 컴포넌트의 명칭을 기재한다.
- 획득 방법
해당 재사용 컴포넌트의 획득 방법을 기재한다. 획득 방법은 기존(legacy), 상용(COTS), 관급(GOTS) 등이며, 재사용 컴포넌트의 획득 방법을 표현하는 적절한 용어를 선택해 기재한다.
- 연계 방법
해당 재사용 컴포넌트의 연계 방법을 기술한다. 재사용 컴포넌트별로 기술하되 인터페이스 또는 인터페이스 오퍼레이션별로 연계 방법이 상이할 경우, 모든 항목에 대해 어떻게 연계할 것인지에 대해 기술한다.
- 부가사항
재사용 컴포넌트를 사용하는데 있어 전제조건, 제약사항 등 부가사항을 기술한다.

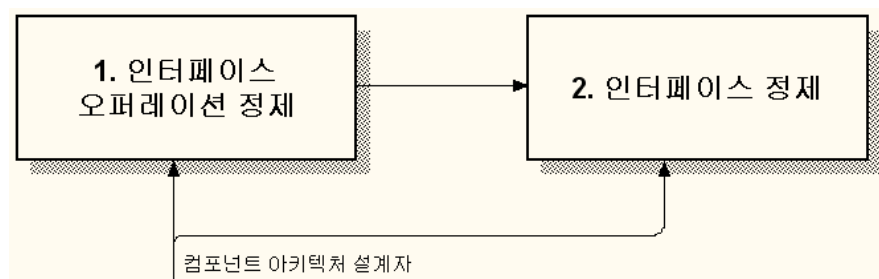
인터페이스 상호작용 명세 작업(2D12)

개요

인터페이스 상호작용 명세 작업은 식별된 컴포넌트 인터페이스 간의 상호작용 분석을 통해 컴포넌트 인터페이스를 정제하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 유스케이스 정의서 클래스 정의서 컴포넌트 아키텍처 명세서
산출물	<ul style="list-style-type: none"> 인터페이스 상호작용 명세서
참여자	<ul style="list-style-type: none"> 컴포넌트 아키텍처 설계자 <p>인터페이스 간 상호작용 분석을 통해 컴포넌트 인터페이스를 정제한다.</p>
관련기법	<ul style="list-style-type: none"> UML

절차 흐름도



절차 설명

1. 인터페이스 오퍼레이션 정제

컴포넌트 식별(2D11)에서 식별된 인터페이스별로 오퍼레이션을 정제한다.

- 인터페이스 오퍼레이션 정제

시퀀스(또는 커뮤니케이션) 다이어그램을 이용해 인터페이스 오퍼레이션의 통합, 분할 및 확장을 수행하고 반환값, 파라미터를 정제한다. 또한, 해당 오퍼레이션을 통해 얻을 수 있는 실행 결과와 사전/사후 조건, 제약사항을 보완해 액터가 서비스를 요청한 후로부터 결과를 받기까지의 일련의 흐름이 실행 가능하고 완전함을 확인한다.

■ 인터페이스 상호작용 명세 작업(2D12)

2. 인터페이스 정제

인터페이스 오퍼레이션 정제 결과를 바탕으로 컴포넌트 인터페이스 및 컴포넌트를 정제하여 컴포넌트를 확정한다.

- 인터페이스 정제

인터페이스 오퍼레이션 정제 결과에 따라 인터페이스의 통합, 분할, 추가, 수정을 수행하고 필요시 컴포넌트의 정제도 수행한다. 인터페이스 정제를 통해 수정/보완된 컴포넌트 인터페이스와 컴포넌트를 기반으로 컴포넌트 아키텍처를 정제한다.

■ 작업 수행시 고려사항

■ 인터페이스 상호작용 명세의 의미

- 인터페이스 상호작용 명세 작업은 컴포넌트 식별(2D11)과 컴포넌트 명세(2D13)를 연결하는 과정이다.
- 컴포넌트 식별(2D11)에서는 컴포넌트를 무엇을 식별할 것인가에 초점을 두어 인터페이스를 식별하였다면 인터페이스 상호작용 명세에서는 실제 인터페이스가 어떻게 활용되어 컴포넌트가 상호작용 하는가에 초점을 두어 인터페이스를 정제한다.
- 시스템의 크기, 복잡도 및 사업의 제약사항에 따라 인터페이스 상호작용 명세 작업은 컴포넌트 식별(2D11)이나 컴포넌트 명세(2D13)와 병행하여 수행할 수 있다. 또한, 컴포넌트 식별(2D11)에서 컴포넌트 간 상호작용을 상세히 수행하였다면, 결과물을 보완하는 방식으로 인터페이스 상호작용 명세 작업을 수행한다.

■ 인터페이스 오퍼레이션 정제, 인터페이스 정제시 고려사항

- 인터페이스와 인터페이스 오퍼레이션을 정제하는 것은 컴포넌트의 경계와 컴포넌트가 제공하는 서비스를 명확히 하는 것을 목적으로 해야 한다. 서비스가 아닌, 인터페이스 자체에 초점을 두어 정제를 수행할 경우 식별된 컴포넌트의 내부 로직까지 접근해 식별된 컴포넌트가 불필요하게 분할될 수 있기 때문이다.
- 컴포넌트는 이미 대체성, 재사용성 및 데이터 응집도를 고려해 식별된 것이므로 컴포넌트를 분할해야 하는 특별한 목적이나 사유 없이 컴포넌트를 변경하지 않도록 주의한다.
- 인터페이스 정제시 컴포넌트 식별(2D11)에서 고려했던 컴포넌트 식별 조건을 다시 한 번 고려해 정제를 수행한다.

■ 인터페이스 상호작용 명세 작업(2D12)

- 인터페이스 상호작용을 모델링 할 때 유스케이스의 사건 흐름 또는 시나리오를 기준으로 하며, 모든 유스케이스가 컴포넌트 간 상호작용을 통해 구현됨이 보장되어야 한다.

■ 인터페이스 상호작용 명세 결과와 이후 작업과의 관계

- 컴포넌트 식별(2D11)과 인터페이스 상호작용 명세(2D12), 컴포넌트 명세(2D13)는 투입 노력과 범위를 고려해 작업이 구분되었으나 작업간의 관련성이 높으므로 3개 작업을 반복 수행하면서 컴포넌트 식별 및 명세의 완전성을 높여야 한다.

■ 인터페이스 상호작용 명세서 표기법

- 인터페이스 상호작용 명세를 위한 표기법은 이후 설계 과정과의 표기법의 일관성을 위해 UML의 시퀀스 또는 커뮤니케이션 다이어그램을 활용하는 것을 원칙으로 하며, 부득이한 경우 인터페이스간의 호출 관계를 표현할 수 있는 다른 표기법을 사용할 수 있다.

■ 인터페이스 상호작용 명세 구성요소간 관계

- 하나의 컴포넌트 인터페이스는 하나 이상의 인터페이스 상호작용으로 표현되어야 한다.

■ 인터페이스 상호작용 명세 작업(2D12)

■ [서식] 인터페이스 상호작용 명세서(2D12a)

1. (유스케이스명)

■ [항목 설명] 인터페이스 상호작용 명세서(2D12a)

1. (유스케이스명)

유스케이스의 사건 흐름 또는 시나리오를 기반으로 컴포넌트와 컴포넌트 인터페이스간의 호출 관계를 다이어그램 또는 다른 방법으로 작성한다.

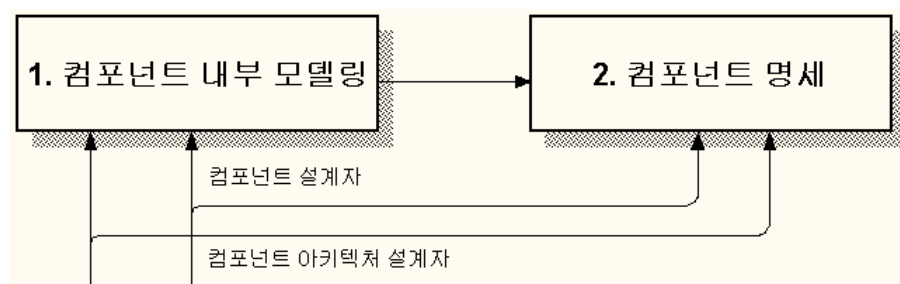
컴포넌트 명세 작업(2D13)

개요

각각의 컴포넌트에 대해 컴포넌트 내부를 모델링하고 컴포넌트와 인터페이스 명세를 작성하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 유스케이스 정의서 • 연동 소요 정의서, (부록)연동 항목 소요서(IER) • 클래스 정의서 • 컴포넌트 아키텍처 명세서 • 인터페이스 상호작용 명세서
산출물	<ul style="list-style-type: none"> • 컴포넌트 명세서
참여자	<ul style="list-style-type: none"> • 컴포넌트 아키텍처 설계자 컴포넌트 아키텍처를 기반으로 컴포넌트 명세를 지원한다. • 컴포넌트 설계자 컴포넌트 내부 로직을 모델링하고 컴포넌트와 인터페이스를 명세한다.
관련기법	<ul style="list-style-type: none"> • UML

절차 흐름도



절차 설명

1. 컴포넌트 내부 모델링

컴포넌트 내부에서 서비스를 처리하는 로직인 클래스를 모델링한다.

컴포넌트 명세 작업(2D13)

- 컴포넌트 내부 클래스 설계
유스케이스 모델링, 클래스 모델링 및 컴포넌트 인터페이스를 고려해 컴포넌트 인터페이스를 구성하는 각 오퍼레이션이 정의된 서비스를 제공하기 위한 내부 클래스를 도출한다. 도출한 내부 클래스에 대해 상세 정의 및 내부 클래스간의 관계를 설계한다.
- 컴포넌트 내부 클래스 모델 정제
컴포넌트 내부 모델링을 수행하면서 컴포넌트 인터페이스부터 컴포넌트 내부 클래스까지의 상호작용을 분석하고 내부 클래스를 정제한다. 또한 컴포넌트 인터페이스에 대한 트랜잭션 참여 여부를 결정한다.
- 알고리즘 설계
내부 클래스의 오퍼레이션 중 복잡한 내부 처리 논리나 업무 규칙이 적용되는 경우 별도 알고리즘을 설계한다.
- 컴포넌트 가변성 설계
유스케이스 모델링에서 식별된 가변성에 대해 가변 요소 및 적용 방안을 식별해 컴포넌트에서 지원해야 할 가변성을 검토하고 구현 관점에서 추가적으로 발생하는 가변성에 대해서도 파악한다. 가변성 구현 방법은 상속 구조 이용, 매개 변수 다형성(메소드 오버로딩), 템플릿 클래스 활용 등 다양한 방법이 있으며, 이중 적합한 방법을 고려해 컴포넌트 내부 클래스를 모델링 한다. 가변성에 따라 인터페이스를 추가/보완한다.

2. 컴포넌트 명세

확정된 컴포넌트 아키텍처 및 컴포넌트를 기반으로 컴포넌트를 명세한다. 컴포넌트 명세는 이전 수행된 최종 결과를 정리하는 것이다.

- 컴포넌트 명세
컴포넌트 내부 모델링 결과를 바탕으로 컴포넌트를 명세한다.

■ 작업 수행시 고려사항

■ 컴포넌트 명세의 의미

- 컴포넌트 명세 작업은 지금까지 수행한 결과를 종합하고 정제하는 작업으로 이후 컴포넌트 구현 설계(2D21)에서는 플랫폼을 고려한 보완과 배포 형태로의 매핑만 수행되므로 컴포넌트 명세 작업이 사실상 컴포넌트에 대한 마지막 설계 작업이다.
- 컴포넌트 기반 개발 방법에 있어서 컴포넌트 명세 결과는 중요도가 높고 컴포넌트 재사용시 활용되므로 형식 위주의 결과물이 되지 않도록 완성도를 높여야 한다.

컴포넌트 명세 작업(2D13)

■ 컴포넌트 내부 모델링시 고려사항

- 컴포넌트 명세는 컴포넌트에 대한 논리적인 설계가 마지막으로 수행되는 작업이다. 따라서 구현 가능한 수준으로 상세한 모델링이 수행되어야 한다.
- 주요 가변성 발생 원인은 다음과 같으며, 컴포넌트 가변성 식별시 고려한다.
 - 사용자의 다양성에 따라 시스템의 기능을 추가하는 경우
 - 관리 대상에 따라 처리 행위는 동일하나 규칙이나 대상이 상이한 경우
 - 군별 특수성에 따라 업무 처리 단계에 대안적/선택적 처리가 필요한 경우
 - 다양한 업무 처리 흐름이나 예외사항 처리 흐름이 필요한 경우
 - 기반 플랫폼에 따라 파라미터의 조정이 필요한 경우
 - 사용자 요구사항 외에도 시스템의 확장성, 이식성을 고려해야 할 경우

■ 컴포넌트 명세 결과와 이후 작업과의 관계

- 컴포넌트 명세(2D13)는 사용자 인터페이스 설계(2D14), 데이터 모델링(2D15)의 결과와 상호 참조하거나 의존하는 항목이 많아 각 작업을 단계별로 수행하는 것은 이후 작업의 변경에 따라 컴포넌트 명세를 재수행 해야 하는 결과가 초래될 수 있으므로 3개 작업을 병행해 수행하는 것이 필요하다.
- 관리적인 측면에서 컴포넌트, 데이터 엔티티, 사용자 화면에 대한 변경 관리가 필요하다.

■ 컴포넌트 명세서 표기법

- 컴포넌트 내부 모델링을 위한 표기법은 이후 설계 과정과의 표기법의 일관성을 위해 UML의 클래스 다이어그램과 시퀀스 또는 커뮤니케이션 다이어그램을 활용하는 것을 원칙으로 한다.

■ 컴포넌트 명세 구성요소간 관계

- 컴포넌트는 없거나 하나 이상의 컴포넌트 가변성을 포함하며, 하나 이상의 컴포넌트 인터페이스를 포함한다. 또한 하나 이상의 컴포넌트 내부 클래스를 포함한다.
- 컴포넌트 인터페이스는 하나 이상의 다른 컴포넌트나 컴포넌트 외부 객체와 호출 관계를 가지며, 하나 이상의 오퍼레이션을 포함한다.
- 컴포넌트 인터페이스 오퍼레이션은 없거나 하나 이상의 오퍼레이션 파라미터를 가지며, 하나 이상의 컴포넌트 내부 클래스와 관계를 가진다.
- 컴포넌트 인터페이스는 파라미터는 없거나 하나 이상의 컴포넌트 가변성과 관계가 있다.

컴포넌트 명세 작업(2D13)

- 컴포넌트 내부 클래스는 하나 이상의 속성과 오퍼레이션을 포함하며, 내부 클래스 오퍼레이션은 없거나 하나 이상의 파라미터를 포함한다. 내부 클래스의 오퍼레이션은 의미적으로 컴포넌트 내부 클래스간 상호작용과 대응된다.

컴포넌트 명세 작업(2D13)

[서식] 컴포넌트 명세서(2D13a)

1. (컴포넌트명)

1.1. 컴포넌트 내부 클래스 다이어그램

1.2. 설명

1.3. 인터페이스

1.3.1. (인터페이스명)

1.3.1.1. 오버레이션

오버레이션 명	사전조건	사후조건	정의 클래스명	설명

1.3.1.2. 내부 상호작용 다이어그램

1.4. 내부 클래스

1.4.1. (클래스명)

1.4.1.1. 속성

속성명	가시성	타입	기본값	설명

1.4.1.2. 오버레이션

오버레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	

1.4.1.3. 알고리즘

■ 컴포넌트 명세 작업(2D13)

1.5. 가변성

1.5.1. (가변성명)

1.5.1.1. 설명

1.5.1.2. 구현 방식

컴포넌트 명세 작업(2D13)

■ [항목 설명] 컴포넌트 명세서(2D13a)

1. (컴포넌트명)

컴포넌트별로 컴포넌트 명세를 작성한다.

1.1. 컴포넌트 내부 클래스 다이어그램

컴포넌트 내부 클래스와 인터페이스 및 컴포넌트와 상호작용하는 외부 객체 또는 컴포넌트 간 관계를 다이어그램으로 작성한다.

1.2. 설명

해당 컴포넌트에 대한 설명을 기술한다. 설명은 컴포넌트의 역할, 주요 기능과 컴포넌트가 관리하는 데이터 및 특징 등을 기술한다.

1.3. 인터페이스

컴포넌트의 인터페이스에 대한 명세를 작성한다. 내부 클래스 다이어그램에 작성 항목이 포함되어 있는 경우 해당 항목을 생략하여 기술할 수 있다.

1.3.1. (인터페이스명)

해당 인터페이스별로 인터페이스 명세를 작성한다.

1.3.2. 오퍼레이션

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.
- 사전조건
해당 오퍼레이션이 설계에 따라 정확히 동작하고 결과를 제공하기 위해 필요한 전제조건 및 가정사항을 기술한다.
- 사후조건
해당 오퍼레이션이 수행된 후 제공되는 결과가 무엇인지 기술한다.
- 정의 클래스명
해당 오퍼레이션이 정의된 클래스의 명칭을 기재한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 작성시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

1.3.3. 내부 상호작용 다이어그램

컴포넌트 인터페이스 오퍼레이션으로부터 내부 클래스까지의 상호작용을 다이어그램으로 작성한다.

컴포넌트 명세 작업(2D13)

1.4. 내부 클래스

컴포넌트 내부의 클래스에 대한 명세를 작성한다. 내부 클래스 다이어그램에 작성 항목이 포함되어 있는 경우 해당 항목을 생략하여 기술할 수 있다.

1.4.1. (클래스명)

클래스별로 클래스 명세를 작성한다.

1.4.1.1. 속성

- 속성명
속성의 명칭을 기재한다.
- 가시성
해당 속성의 가시성을 기재한다.
- 타입
해당 속성의 데이터 타입을 기재한다.
- 기본값
객체 생성시 해당 속성의 기본값을 기재한다.
- 설명
속성에 대한 설명을 기술한다. 설명 기술시 '로그인 화면의 비밀번호', '부대부호를 임시로 저장하기 위한 속성' 등 속성이 의미하는 것에 초점을 두어 기술한다.

1.4.1.2. 오퍼레이션

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.
- 가시성
해당 오퍼레이션의 가시성을 기재한다.
- 반환 타입
해당 오퍼레이션의 반환 데이터 타입을 기재한다.
- 파라미터 명칭
오퍼레이션 파라미터의 명칭과 데이터 타입을 기재한다.
- 파라미터 설명
오퍼레이션 파라미터의 설명을 기술한다. 파라미터 설명 기술시 해당 파라미터가 의미하는 것에 초점을 두어 기술한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 기술시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

컴포넌트 명세 작업(2D13)

1.4.1.3. 알고리즘

해당 클래스의 주요 알고리즘을 기술한다.

1.5. 가변성

컴포넌트의 가변성을 명세한다.

1.5.1. (가변성명)

가변성별로 작성한다.

1.5.1.1. 설명

해당 가변성을 설계하게 된 원인과 가변성을 통해 얻어지는 결과 및 가변성에 따라 영향을 받는 대상 등을 기술한다.

1.5.1.2. 구현 방식

가변성을 구현한 방식을 컴포넌트 내부 클래스 및 인터페이스와 연관지어 기술한다.

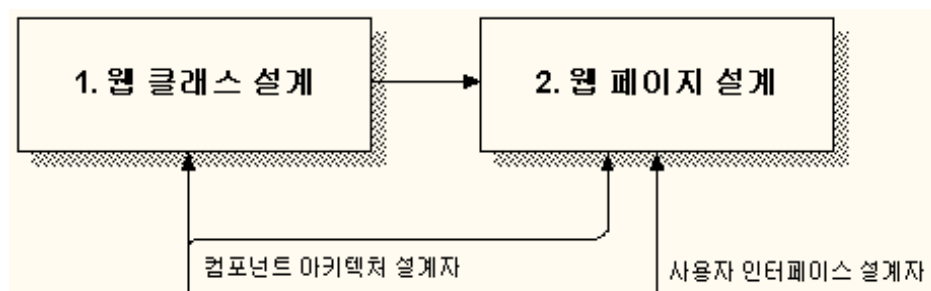
■ 사용자 인터페이스 명세 작업(2D14)

■ 개 요

사용자 인터페이스 정의 결과를 바탕으로 웹 페이지와 사용자 인터페이스 제어 역할을 하는 클래스를 설계하고 웹 페이지 및 웹 클래스간의 관계를 정의하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 시스템 아키텍처 정의서 • 유스케이스 정의서 • 클래스 정의서 • 사용자 인터페이스 정의서 • 컴포넌트 아키텍처 명세서 • 컴포넌트 명세서
산출물	<ul style="list-style-type: none"> • 사용자 인터페이스 명세서
참여자	<ul style="list-style-type: none"> • 컴포넌트 아키텍처 설계자 컴포넌트 아키텍처 및 컴포넌트 설계 결과를 기반으로 웹 클래스를 설계하고 웹 페이지와 웹 클래스 간의 관계를 설계한다. • 사용자 인터페이스 설계자 웹 페이지를 설계하고 웹 페이지 및 웹 클래스 간의 관계를 설계한다.
관련기법	<ul style="list-style-type: none"> • UML

■ 절차 흐름도



■ 절차 설명

1. 웹 클래스 설계

웹 페이지 호출 관계를 제어하거나 공통적으로 사용되는 웹 클래스를 설계한다.

■ 사용자 인터페이스 명세 작업(2D14)

- 웹 페이지 식별
정의된 사용자 인터페이스를 기반으로 웹 페이지를 식별한다.
- 웹 클래스 설계
웹 페이지의 복잡한 호출 관계를 제어하는 클래스, 공통적으로 사용되거나 빈번히 사용되는 클래스, 성능상의 이유로 필요하다고 판단되는 클래스들이 존재할 경우 이들 클래스를 식별하고 각 클래스의 오퍼레이션, 속성을 설계한다.
- 웹 컴포넌트 인터페이스 식별
그래프, 그리드 등 화면 구성을 위한 UI 컴포넌트와 리포팅 도구 등 웹 컴포넌트 및 이를 활용하기 위한 인터페이스(API)를 식별한다.

2. 웹 페이지 설계

웹 페이지의 공통 템플릿을 설계하고 이 템플릿을 이용하여 웹 페이지를 설계한다.

- 사용자 인터페이스 템플릿 설계
사용자 인터페이스에 대한 정의 결과, 표준 지침을 기반으로 사용자 인터페이스에 대한 공통 템플릿을 설계한다.
- 웹 구성요소간의 상호작용 관계 설계
웹 페이지, 웹 콘텐츠(이미지, 사운드 등)를 정의하고 웹 클래스, 웹 페이지, 웹 콘텐츠 및 웹 컴포넌트 등 웹 구성요소간의 상호작용 관계를 고려하여 이들 간의 호출 흐름을 설계한다.

■ 작업 수행시 고려사항

■ 사용자 인터페이스 명세의 의미

- 사용자 인터페이스 명세는 앞서 정의된 화면, 보고서 등을 어떻게 시스템으로 구현할 것인가에 초점을 둔다. 특히 시스템 아키텍처 및 컴포넌트 아키텍처에 따라 사용자의 서비스 요청이 어떤 호출 관계를 통해 비즈니스 컴포넌트로 전달될 것인지를 식별하는 것이 중요하다.
- 사용자 인터페이스 명세는 특정 플랫폼이나 기술을 고려하지 않으나 사용자 인터페이스의 특성상 EJB, .NET 등 적용 플랫폼이나 사용하는 기술에 따라 웹 페이지 및 웹 클래스의 구성이 변경될 수 있으므로 본 작업을 상세 설계(2D2) 활동의 사용자 인터페이스 구현 설계(2D22)와 함께 수행하는 것을 고려할 수 있으며, 이 경우 다른 작업들 간의 관계를 고려해 방법론을 테일러링한다.

사용자 인터페이스 명세 작업(2D14)

■ 웹 클래스 설계시 고려사항

- 웹 클래스가 웹 페이지와 비즈니스 컴포넌트간의 중계 역할에 초점을 두고 웹 컴포넌트의 인터페이스 사용 방법, 비즈니스 컴포넌트 호출 방법 등 웹 클래스의 내부 로직이 패턴을 가지며 웹 클래스의 내부 로직이 표준 지침 수립(1R22a)의 코딩 표준 등으로 제시될 경우, 웹 클래스 설계에서는 별도의 설계 과정을 생략하고 호출 관계 식별을 위한 웹 클래스의 오퍼레이션만 설계해도 무방하다.

■ 웹 페이지 설계시 고려사항

- 웹 구성요소간의 호출 관계는 정적/동적 관계를 모두 고려해 설계되어야 사용자 인터페이스의 구현 및 관리가 용이하다.
- 웹 페이지간의 호출 관계는 다음의 사항을 기준으로 웹 페이지 식별 결과에 따라 확장한다.

호출 관계 구분	설 명
Build	다른 웹 페이지로부터 생성되는 것이 아닌 단일 HTML 또는 JSP/ASP 등 서버 페이지로 구현할 웹 페이지
Link	해당 웹 페이지와 단순 하이퍼링크 관계를 갖는 웹 페이지 ※ 메인 또는 공통 페이지 등에 대해서는 작성하지 않아도 무방하며, 다른 시스템 또는 웹 포털 등과 관계가 있는 경우 기술한다.
Include	프레임 내에 포함된 웹 페이지 또는 호출한 웹 페이지가 갖고 있는 정보로 생성되며 자체적으로 웹 클래스 호출이 없는 팝업 페이지

■ 사용자 인터페이스 명세 결과와 이후 작업과의 관계

- 사용자 인터페이스 명세(2D14)에서 정의된 웹 구성요소를 기반으로 사용자 인터페이스 구현 설계(2D22)에서 실 운용환경의 하드웨어에 해당 파일들을 배치하기 위한 설계가 수행된다.

■ 웹 페이지와 클래스간의 관계 설계시 표기법

- 웹 페이지, 웹 클래스들 간의 호출 관계만 식별된다면 호출 관계를 스프레드시트로 정리해도 무방하다. 그러나 호출 관계에 대한 가독성과 이해성을 높이고 전체적인 호출 관계를 파악하기 위해서는 도형화된 표기법을 사용하는 것이 좋다.

■ 사용자 인터페이스 명세 작업(2D14)

- 호출 관계에 대한 정적인 구조는 UML의 클래스 다이어그램을 적용할 수 있으며, 동적인 구조는 UML의 시퀀스 또는 커뮤니케이션 다이어그램을 적용할 수 있다. 또한 웹 구성을 표현하기 위한 특화된 표기법이나 확장된 UML 표기법도 활용할 수 있다.

■ 사용자 인터페이스 명세 구성요소간 관계

- 웹 페이지는 여러 개의 웹 콘텐츠와 관계가 있으며, 웹 콘텐츠도 여러 개의 웹 페이지와 관계를 가진다. 웹 페이지, 웹 클래스, 웹 컴포넌트는 하나 이상의 상호 호출 관계를 가진다.
- 웹 클래스는 하나 이상의 오퍼레이션과 속성을 가지며, 오퍼레이션은 하나 이상의 파라미터를 포함한다.

■ 사용자 인터페이스 명세 작업(2D14)

■ [서식] 사용자 인터페이스 명세서(2D14a)

1. 웹 구성 다이어그램

2. 웹 클래스

2.1. 웹 클래스 목록

웹 클래스명	Submit	Forward	Server	설명

2.2. (클래스명)

2.2.1. 속성

속성명	가시성	타입	기본값	설명

2.2.2. 오퍼레이션

오퍼레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	

2.2.3. 알고리즘

3. 웹 컴포넌트

3.1. 웹 컴포넌트 목록

웹 컴포넌트명	설명	인터페이스		
		명칭	오퍼레이션명	오퍼레이션설 명

■ 사용자 인터페이스 명세 작업(2D14)

4. 웹 페이지

4.1. 웹 페이지 목록

웹 페이지명	설명	주요 웹 콘텐츠	관련 페이지	
			명칭	관계

■ 사용자 인터페이스 명세 작업(2D14)

■ [항목 설명] 사용자 인터페이스 명세서(2D14a)

1. 웹 구성 다이어그램

웹 페이지, 웹 클래스 및 웹 컴포넌트간의 관계를 다이어그램으로 작성한다. 정적 관계는 반드시 작성되어야 하며, 호출 관계가 복잡할 경우 동적 관계도 작성한다.

2. 웹 클래스

웹 클래스의 관계와 오퍼레이션을 중심으로 웹 클래스 설계 결과를 작성한다. 웹 구성 다이어그램에 작성 항목이 포함되어 있는 경우 해당 항목을 생략해 기술할 수 있다.

2.1. 웹 클래스 목록

웹 클래스 목록을 웹 페이지 등 타 구성요소와의 관계를 고려해 작성한다.

- 웹 클래스명
웹 클래스의 명칭을 기재한다.
- Submit
Submit을 통해 서비스를 요청하는 웹 페이지명을 기재한다.
- Forward
웹 클래스에서 Forwarding하는 웹 페이지를 기재한다.
- Server
해당 웹 클래스가 호출하는 서버측의 비즈니스 컴포넌트 인터페이스를 기재한다.
- 설명
웹 클래스가 중계하는 주요 기능을 중심으로 설명을 기술한다.

2.2. (클래스명)

해당 클래스별로 작성한다.

2.2.1. 속성

- 속성명
속성의 명칭을 기재한다.
- 가시성
해당 속성의 가시성을 기재한다.
- 타입
해당 속성의 데이터 타입을 기재한다.
- 기본값
객체 생성시 해당 속성의 기본값을 기재한다.

사용자 인터페이스 명세 작업(2D14)

- 설명
속성에 대한 설명을 기술한다. 설명 기술시 '로그인 화면의 비밀번호', '부대부호를 임시로 저장하기 위한 속성' 등 속성이 의미하는 것에 초점을 두어 기술한다.

2.2.2. 오퍼레이션

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.
- 가시성
해당 오퍼레이션의 가시성을 기재한다.
- 반환 타입
해당 오퍼레이션의 반환 데이터 타입을 기재한다.
- 파라미터 명칭
오퍼레이션 파라미터의 명칭과 데이터 타입을 기재한다.
- 파라미터 설명
오퍼레이션 파라미터의 설명을 기술한다. 파라미터 설명 기술시 해당 파라미터가 의미하는 것에 초점을 두어 기술한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 기술시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

2.2.3. 알고리즘

해당 클래스의 주요 알고리즘을 기술한다.

3. 웹 컴포넌트

3.1. 웹 컴포넌트 목록

웹 컴포넌트 목록을 컴포넌트가 제공하는 인터페이스에 초점을 두어 작성한다. 웹 구성 다이어그램에 작성 항목이 포함되어 있는 경우 해당 항목을 생략해 기술할 수 있다.

- 웹 컴포넌트명
웹 컴포넌트의 명칭을 기재한다.
- 설명
웹 컴포넌트가 제공하는 기능, 웹 컴포넌트의 특징 등을 기술한다.

■ 사용자 인터페이스 명세 작업(2D14)

- 인터페이스 명칭
웹 컴포넌트가 제공하는 인터페이스 명칭을 기재한다.
- 오퍼레이션명
해당 인터페이스에서 제공하는 오퍼레이션 명칭을 기재한다.
- 오퍼레이션 설명
해당 오퍼레이션의 주요 파라미터와 반환값, 주요 기능을 기술한다.

4. 웹 페이지

4.1. 웹 페이지 목록

웹 페이지 목록을 웹 페이지 간 관계에 초점을 두어 작성한다. 웹 구성 다이어그램에 작성 항목이 포함되어 있는 경우 해당 항목을 생략하여 기술할 수 있다.

- 웹 페이지명
웹 페이지의 명칭을 기재한다.
- 설명
웹 페이지가 표현하는 화면명, 주요 정보를 기술한다.
- 주요 웹 콘텐츠
해당 웹 페이지에 포함된 주요 웹 콘텐츠를 기재한다. 주요 웹 콘텐츠는 업무 수행에 관련 있는 웹 콘텐츠로 한정한다.
- 관련 페이지 명칭
해당 웹 페이지와 관계를 갖고 있는 웹 페이지 명칭을 기재한다.
- 관련 페이지 관계
관계가 있는 웹 페이지와 어떤 관계가 있는지를 기술한다.

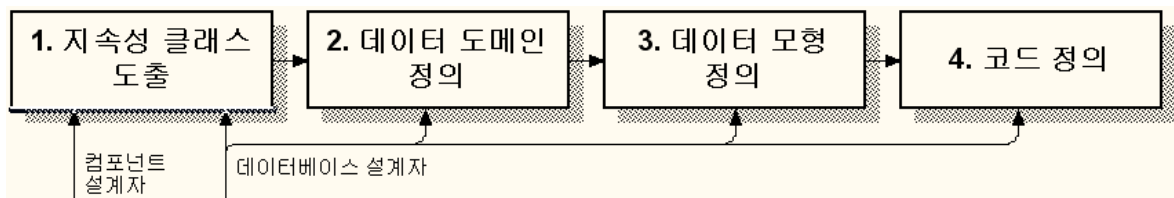
데이터 모델링 작업(2D15)

개요

시스템에서 지속적으로 관리되어야 하는 데이터와 그 특성을 식별하여 개발자 간 공유하기 위해 데이터 모형을 모델링하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 클래스 정의서 연동 소요 정의서 자료 구축 계획서
산출물	<ul style="list-style-type: none"> 데이터 명세서
참여자	<ul style="list-style-type: none"> 데이터베이스 설계자 <p>지속성 클래스를 기반으로 데이터 모형과 공통성을 정의한다.</p>
관련기법	<ul style="list-style-type: none"> 데이터 모델링
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 데이터공유환경(SHADE) 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 지속성 클래스 도출

클래스 모델링 결과를 기반으로 지속성 클래스를 도출한다.

- 지속성 클래스 도출

클래스 다이어그램에서 지속적으로 정보를 관리해야 하는 지속성 클래스를 찾아 도출한다.

데이터 모델링 작업(2D15)

2. 데이터 도메인 정의

논리 엔티티에서 적용해야 할 데이터 도메인을 정의한다. 데이터 도메인 정의는 데이터 모형 정의와 분리하여 실시하는 것이 아니라 데이터 모형을 정의하면서 도메인을 검색하여 반영하고, 추가적으로 정의해야 할 필요가 식별되었을 경우 이를 추가하는 과정을 반복하는 것이다.

- 초기 데이터 도메인 정의
식별된 지속성 클래스의 속성을 분석해 엔티티 속성의 일반적인 특징을 나타내는 데이터의 성격을 분류하여 초기 데이터 도메인을 정의하고 공용으로 선언한다.
- 데이터 도메인 정제
데이터베이스 설계자가 데이터 모형을 정의하면서 도메인을 정제한다.

3. 데이터 모형 정의

논리 엔티티를 정의하고 엔티티간의 관계를 모델링한다.

- 논리 엔티티 식별
지속성 클래스를 기반으로 논리 엔티티를 식별한다. 지속성 클래스로는 도출되지 않았으나 업무를 지원하기 위해 생성되어야 하는 엔티티도 논리 엔티티로 식별한다.
- 논리 엔티티 정의
도출한 엔티티의 특성이나 속성을 파악하고 클래스 모델링 결과와 지속성 클래스간의 연관 관계를 고려하여 엔티티간 관계성을 설계한다. 지속성 클래스는 하나 이상의 논리 엔티티로 매핑이 가능하지만, 데이터 접근 성능을 고려하여 상속관계 매핑, 연관관계 매핑 등 여러 가지 매핑 전략이 사용될 수 있다.
- 데이터 모형 정제
성능 향상, 데이터 무결성 및 적합성 향상을 위해 정규화 작업을 수행한다.

■ 작업 수행시 고려사항

■ 데이터 모델링 작업의 수행 여부

- 데이터 모델링 작업은 소규모 사업의 경우 데이터베이스 설계 작업과 병행하고 산출물로 데이터베이스 설계서만 작성할 수 있다.
- 객체지향 데이터베이스 시스템을 사용한다면 본 작업의 절차에 준해 데이터 모형을 객체지향 관점으로 작성한다.

데이터 모델링 작업(2D15)

■ 데이터 도메인 정의시 고려사항

- 데이터 도메인 정의 시 국방 표준을 준수해야 하므로, 국방메타데이터관리체계에서 제시하고 있는 ‘데이터 표준화 지침서’에 따라 도메인 정의 절차를 수행하고 시스템에 반영해야 한다.

■ 데이터 모형 정의시 고려사항

- 엔티티 및 속성 등 데이터 요소에 대한 명명규칙은 ‘국방상호운용성관리 지침서’에 의거하여 국방메타데이터관리체계에서 제시하고 있는 ‘데이터 표준화 지침서’의 명명규칙을 준수한다. 즉, 표준 단어와 표준 도메인을 활용하여 데이터 요소를 정의하고, 표준 단어나 표준 도메인에 대한 등록 또는 수정이 필요한 경우 변경 요청한다.
- 또한, 데이터 모형 정의 시 국방메타데이터관리체계에서 관리하고 있는 국방 데이터 모델을 참조하여야 한다.

■ 데이터 속성 명명시의 유의사항

- 데이터의 속성명이 이해성이 높은지, 프로그래밍 언어에 사용됨에 있어서 적절한지, 일관성이 높은지에 대해 유의해야 한다.
- 예를 들어 ‘일자1’, ‘일자2’ 등으로 의미를 구분하기 어려운 속성은 부득이한 경우가 아니면 ‘초도 지원 일자’, ‘후속 지원 일자’와 같이 이해도를 높여야 한다. 단순히 ‘구분’과 같이 의미를 이해하기 어려운 속성명도 개선해야 한다.
- 속성명이 너무 길어 가독성이 떨어지고 코딩하기에도 불편한 경우가 없도록 해야 한다.
- 사용자 화면 및 보고서의 각 항목 명칭과 논리 엔티티의 속성명이 일치하지 않은 경우는 화면의 내용이 어느 엔티티의 속성에서 온 것인지 알 수가 없으므로 화면, 보고서의 항목명과 논리 엔티티의 속성명을 가급적 고객이 사용하는 표준 용어로 일치시켜야 한다.

■ 데이터 속성 설계시 유의사항

- 데이터의 속성에서 동음이의(동일한 명칭이나 다른 의미를 가짐), 이음동의(다른 명칭이 부여되어 있으나 동일한 대상임), 동음동의 이형(명칭은 일관성 있게 부여되어 있으나 데이터 타입과 또는 길이 등이 다르게 정의됨) 현상이 발생하지 않도록 해야 한다.

데이터 모델링 작업(2D15)

■ 모델링 파일 제출

- 일반적으로 모델링 도구를 이용하여 데이터를 설계하므로 세부 정보 확인, 형상 관리를 위해 데이터 명세서외에도 모델링 파일이나 기타 관련 파일을 함께 산출물로 제출해야 한다.

■ 데이터 모델링 결과와 이후 작업과의 관계

- 이후 작업과의 관계는 다음과 같다. 논리 데이터 모형은 데이터베이스 설계 작업(2D23)의 물리 데이터 모형으로, 엔티티는 테이블로, 속성은 컬럼으로 유도된다.

데이터 모델링 작업(2D15)

■ [서식] 데이터 명세서(2D15a)

1. 논리 데이터 모형

■ [항목 설명] 데이터 명세서(2D15a)

1. 논리 데이터 모형

논리 ER 다이어그램으로 작성하되, 엔티티, 속성, 논리 데이터 타입 및 길이, 도메인명이 식별될 수 있도록 작성한다.

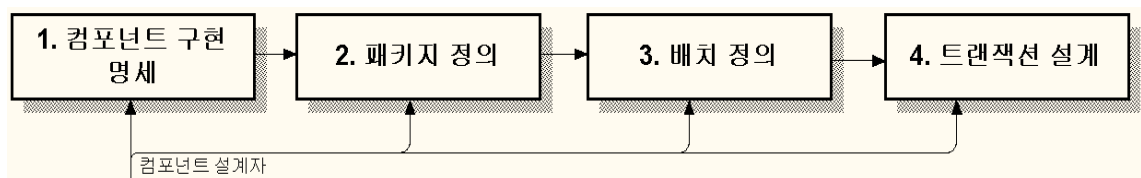
컴포넌트 구현 설계 작업(2D21)

개요

컴포넌트 명세 결과를 기반으로 각 컴포넌트를 구현 플랫폼에 맞게 설계한다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 유스케이스 정의서 클래스 정의서 컴포넌트 아키텍처 명세서 인터페이스 상호작용 명세서 컴포넌트 명세서
산출물	<ul style="list-style-type: none"> 컴포넌트 설계서 트랜잭션 설계서
참여자	<ul style="list-style-type: none"> 컴포넌트 설계자 <p>컴포넌트 명세 결과를 개발 플랫폼을 고려해 정제하고 컴포넌트의 배포 및 배치를 정의한다.</p>
관련기법	<ul style="list-style-type: none"> UML

절차 흐름도



절차 설명

1. 컴포넌트 구현 명세

컴포넌트를 구성하는 내부 클래스를 구현 플랫폼에 맞춰 보완한다.

- 컴포넌트 내부 클래스 보완

컴포넌트 내부 클래스를 구현 플랫폼에 맞게 상세 설계하여 클래스 다이어그램으로 작성하고 인터페이스의 오퍼레이션을 정의한다. 여기서 인터페이스의 오퍼레이션은 구현 플랫폼에 맞는 상세한 시그니처를 포함한 오퍼레이션이다. 컴포넌트의 내부 클래스 및 컴포넌트 인터페이스가 어떻게 구현 클래스로 매핑되는지를 정의하고 클래스의 성격에 따라 클래스의 매핑 타입을 기술한다.

컴포넌트 구현 설계 작업(2D21)

- 기타 클래스 설계
플랫폼 종속에 따라 추가적으로 구현해야 하는 클래스를 설계한다.

2. 패키지 정의

유스케이스 모델링 결과와 컴포넌트 명세 결과를 기반으로 컴포넌트가 구현되는 플랫폼 환경에서 각 구현 클래스가 존재하게 될 위치를 정의한다. 패키지는 개발 과정에서 프로그램 소스의 보관 및 배포의 단위로 사용하고 개발 과정의 각종 관리 작업 단위로 사용한다. 따라서 패키지 도출 기준은 해당 프로젝트 팀의 조직, 서브시스템 구성, 아키텍처 구조에 준하여 정의하여야 한다.

- 시스템 단위 패키지 정의
시스템 패키지 정보는 유스케이스 정의서를 통해서 추출한다. 추출한 정보는 시스템 단위 패키지화에 기초 정보가 된다.
ex) package status (J2EE), namespace Status (.NET)//시스템 단위
- 컴포넌트 단위 패키지 정의
컴포넌트 명세서를 바탕으로 각 업무 별로 추출된 컴포넌트 이름을 이용할 수 있다. 컴포넌트 이름을 이용하여 시스템 단위 패키지 이름의 하위 구조로 각 컴포넌트 별 패키지 구조를 정의한다.
ex) package status.overlaycomp(J2EE),
namespace Status.OverLayComp(.NET)//시스템 아래의 컴포넌트 단위
- 컴포넌트 단위 패키지의 하위 패키지 정의
실제 구현할 클래스를 패키지 구조로 나누어 관리하는 단계로서 각 클래스의 속성을 고려하여 패키지 구조로 정의한다.
ex) package status.overlaycomp.graph(J2EE),
namespace Status.OverLayComp.Graph(.NET);//컴포넌트 아래에 해당 resource를 특성에 따라 구분

3. 배치 정의

컴포넌트의 배포 단위를 정의하고 하나의 배포 단위 내에 존재하는 여러 개의 컴포넌트와 클래스 등 배포 구성 요소를 정의한다. 또한 시스템의 각 하드웨어 구성 요소 별로 컴포넌트를 할당하여 디플로이먼트 다이어그램으로 나타낸다.

- 배포 정의
배포 단위는 한 어플리케이션 전체를 대상으로 구성할 수도 있고, 부분적으로 구성할 수도 있다. 기능적인 측면에서 서로 유기적으로 묶이는 단위로 구성된

컴포넌트 구현 설계 작업(2D21)

논리적인 단위의 컴포넌트를 배포 단위로 구성하는 것이 합리적인 재사용 배포 단위가 될 수 있다.

- 컴포넌트 배치

시스템을 구성하는 구성 요소를 정의하고 시스템 구성 요소와 구성 요소에 존재하는 컴포넌트를 표현하여 물리적인 하드웨어와 개발된 소프트웨어와의 배치 상태를 표현한다.

4. 트랜잭션 설계

유스케이스 모델을 기반으로 트랜잭션을 정의한다. 트랜잭션은 통상적으로 데이터베이스에 접근하는 일의 단위를 수행하는 것을 말한다. 여기서 일의 단위란 상호 연관되고 함께 완료되어야 하는 행위들의 집합이다.

- 트랜잭션 설계

컴포넌트별로 유스케이스 기반의 트랜잭션에 참여하는 컴포넌트의 인터페이스 오퍼레이션을 파악하여 트랜잭션 목록을 기술하고 트랜잭션 종류 및 격리 수준을 명세한다. 또한, 예외 상황이 발생했을 때 트랜잭션의 진행 여부를 결정하고 예외 처리 및 롤백(Roll-back)을 위한 예외 상황별 예외 처리 지침을 정의한다.

■ 작업 수행시 고려사항

■ 컴포넌트 구현 설계의 의미

- 개략 설계(2D1)의 컴포넌트 명세(2D13)에서 컴포넌트 내부의 로직 관점에서 컴포넌트를 명세하였다면 컴포넌트 구현 설계는 실제 구현된 컴포넌트의 모습을 설계하는 작업으로 플랫폼에 독립적으로 설계해 온 컴포넌트를 구현 관점에서 플랫폼에서 지원하는 매커니즘으로 정제하는 작업이다. 또한, 개발된 컴포넌트가 배치될 물리적인 위치를 결정한다.

■ 컴포넌트 구현 명세서 고려사항

- 현재 J2EE, .NET이 가장 일반적으로 적용되는 플랫폼이며, 이에 따라 컴포넌트를 적용하려는 플랫폼의 매커니즘으로 정확히 변환하는 것이 필요하다.
- 예를 들어 J2EE를 적용할 경우 빈의 종류(세션 빈, 엔티티 빈, 메시지드리븐 빈) 및 클래스의 종류(VO, DAO 등) 등으로 클래스의 정제를 고려하고 컴포넌트 인터페이스에 대해 홈/리모트 인터페이스를 정의해야 한다. .NET을 적용할 경우 인터페이스, 서비스드 컴포넌트(Serviced-Component), 데이터셋(DataSet) 등을 고려해야 한다.

컴포넌트 구현 설계 작업(2D21)

- 프로젝트 요구사항 중 특정 플랫폼의 적용이 명시되어 있거나 프레임워크를 사용해 구현 플랫폼으로의 종속을 제거할 경우 컴포넌트 명세(2D13)와 컴포넌트 구현 설계(2D21)의 작업을 조정해 중복 작업을 최소화하도록 한다.
- 구현 플랫폼을 적용함에 있어 컴포넌트 명세의 내부 클래스가 변경되거나 통합되어 구현 클래스와의 직관적인 매핑이 곤란할 경우 별도의 구현 클래스 설계 및 상호작용 설계 절차를 수행해야 하며, 작업의 결과물로 관리해 운용 및 유지보수에서 활용할 수 있도록 한다.
- 구현 플랫폼으로의 종속을 위해 컴포넌트 인터페이스와 내부 클래스를 정제하는 활동이 패턴화 가능한 경우 표준 지침으로 작성해 관리한다.

■ 트랜잭션 설계시 고려사항

- J2EE와 .NET에서 명시하는 트랜잭션의 종류와 격리수준은 다음과 같다.
 - NotSupported : 자신을 호출한 컴포넌트가 트랜잭션을 지원하는 것과 상관없이 트랜잭션 처리를 하지 않고 수행한다.
 - Supports : 자신을 호출한 컴포넌트가 트랜잭션을 지원하거나 새로 트랜잭션을 생성했을 경우, 해당 트랜잭션의 바운더리에 참여하게 된다.(.NET의 Supported와 동일)
 - Required : 자신을 호출한 컴포넌트가 트랜잭션을 지원하거나 새로 트랜잭션을 생성했을 경우, 해당 트랜잭션의 바운더리에 참여하게 되며, 자신을 호출한 컴포넌트가 트랜잭션이 없는 경우 자신의 트랜잭션 바운더리를 새로 생성한다.
 - RequiresNew : 자신을 호출한 컴포넌트가 트랜잭션을 지원하는 것과 상관없이 항상 자신의 트랜잭션 바운더리를 새로 생성한다.
 - Mandatory : 어떤 컴포넌트가 반드시 자신을 호출한 컴포넌트의 트랜잭션 범위의 일부가 되어야 한다. 그렇지 않으면 예외를 리턴하고 실패한다. (J2EE에서만 적용)
 - Never : 컴포넌트가 트랜잭션 영역하에 호출되지 않음을 의미한다. 만약 호출하는 컴포넌트가 트랜잭션 중에 있을 경우 예외를 리턴한다. (J2EE에서만 적용)
 - UserManaged : 트랜잭션을 프로그램 코딩으로 설정한다. (J2EE에서만 적용)
 - Disabled : 트랜잭션을 설정할 필요가 없고, 자신을 호출한 컴포넌트의 트랜잭션을 무시한다. (.NET에서만 적용)

■ 컴포넌트 구현 설계 작업(2D21)

- 트랜잭션 격리 수준은 다음의 항목을 참고한다.
 - ReadCommitted : 트랜잭션은 커밋되지 않은 데이터를 읽지 못한다. 즉 다른 트랜잭션에 의해 변경 중인 데이터는 읽을 수 없다.
 - ReadUnCommitted : 트랜잭션은 커밋되지 않은 데이터를 읽을 수 있다. 이 격리 수준에서 클래스는 커밋되지 않은 변경 사항을 읽을 수 있다.
 - RepeatableRead : 트랜잭션은 다른 트랜잭션에 의해 읽혀지고 있는 데이터를 변경할 수 있다.
 - Serializable : 데이터에 대한 배타적인 읽기와 쓰기 권한을 가진다. 다른 트랜잭션은 그 데이터를 읽거나 쓸 수 없다. 이 격리 수준은 가장 제약이 많다.

■ 컴포넌트 구현 설계 결과와 이후 작업과의 관계

- 컴포넌트 구현 설계 결과를 기반으로 사용자 인터페이스 구현 설계(2D22)에서 웹 패키지와의 관계 및 배치를 결정하게 되며, 코딩시 컴포넌트 구현 설계 결과를 기반으로 개발자들이 컴포넌트를 구현하게 된다.

■ 컴포넌트 구현 설계시 표기법

- 구현 플랫폼으로의 매핑은 컴포넌트 명세시와 마찬가지로 UML의 클래스 다이어그램과 상호작용 설계를 위한 시퀀스 또는 커뮤니케이션 다이어그램을 사용한다.
- 컴포넌트 배치를 표현할 때에는 UML의 디플로이먼트 다이어그램 표기법을 적용하며, 시스템 구성 요소와 구성 요소에 존재하는 컴포넌트를 표현할 수 있는 표기법이 있다면 다른 표기법을 사용할 수 있다.

■ 컴포넌트 구현 설계 구성 요소간 관계

- 컴포넌트 명세(2D13)의 내부 클래스와 컴포넌트 구현 설계(2D21)의 구현 클래스는 다대다 관계를 가진다.
- 컴포넌트는 하나의 패키지에 포함되며, 패키지는 하나 이상의 컴포넌트를 포함한다. 컴포넌트는 배포 파일을 통해 시스템의 하드웨어와 관계를 가진다.

컴포넌트 구현 설계 작업(2D21)

[서식] 컴포넌트 설계서(2D21a)

1. (컴포넌트명)

1.1. 구현 클래스

1.1.1. 구현 클래스 매핑 목록

내부 클래스명	매핑 타입	구현 클래스명	설명

1.1.2. (클래스명)

1.1.2.1. 속성

속성명	가시성	타입	기본값	설명

1.1.2.2. 오퍼레이션

오퍼레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	

1.2. 인터페이스

1.2.1. (인터페이스명)

1.2.1.1. 오퍼레이션

오퍼레이션 명	사전조건	사후조건	정의 클래스명	설명

2. 기타 클래스

2.1. (클래스명)

2.1.1. 설명

컴포넌트 구현 설계 작업(2D21)

2.1.2. 속성

속성명	가시성	타입	기본값	설명

2.1.3. 오퍼레이션

오퍼레이션 명	가시성	반환 타입	파라미터		설명
			명칭	설명	

2.1.4. 알고리즘

3. 패키지 정의

패키지명	클래스명	설명

4. 배치 정의

4.1. 디플로이먼트 다이어그램

4.2. 배포 정의

배포 파일명	컴포넌트명	클래스명	설명

컴포넌트 구현 설계 작업(2D21)

■ [항목 설명] 컴포넌트 설계서(2D21a)

1. (컴포넌트명)

해당 컴포넌트별로 명세를 작성한다.

1.1. 구현 클래스

컴포넌트의 구현 클래스에 대한 명세를 작성한다.

1.1.1. 구현 클래스 매핑 목록

컴포넌트 명세의 내부 클래스와 구현 클래스간의 매핑 관계를 작성한다.

- 내부 클래스명
매핑 대상인 내부 클래스의 명칭을 기재한다.
- 매핑 타입
컴포넌트의 내부 클래스를 구현 클래스로 매핑하기 위한 플랫폼 종속적인 타입을 기술한다.
- 구현 클래스명
내부 클래스와 매핑되는 구현 클래스의 명칭을 기재한다.
- 설명
구현 클래스로의 매핑에 따른 매핑 기준이나 주요 변경 내역 등 추가적인 내용을 기술한다.

1.1.2. (클래스명)

해당 클래스별로 클래스 명세를 작성한다. 내부 클래스와 구현 클래스간의 매핑이 표준화된 패턴에 따르며, 매핑에 따른 속성이나 오퍼레이션의 변화가 없을 경우 생략할 수 있다.

1.1.2.1. 속성

- 속성명
속성의 명칭을 기재한다.
- 가시성
해당 속성의 가시성을 기재한다.
- 타입
해당 속성의 데이터 타입을 기재한다.
- 기본값
객체 생성시 해당 속성의 기본값을 기재한다.

컴포넌트 구현 설계 작업(2D21)

- 설명
속성에 대한 설명을 기술한다. 설명 기술시 '로그인 화면의 비밀번호', '부대부호를 임시로 저장하기 위한 속성' 등 속성이 의미하는 것에 초점을 두어 기술한다.

1.1.2.2. 오퍼레이션

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.
- 가시성
해당 오퍼레이션의 가시성을 기재한다.
- 반환 타입
해당 오퍼레이션의 반환 데이터 타입을 기재한다.
- 파라미터 명칭
오퍼레이션 파라미터의 명칭과 데이터 타입을 기재한다.
- 파라미터 설명
오퍼레이션 파라미터의 설명을 기술한다. 파라미터 설명 기술시 해당 파라미터가 의미하는 것에 초점을 두어 기술한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 기술시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

1.2. 인터페이스

컴포넌트의 인터페이스에 대한 명세를 작성한다. 구현 매핑에 따라 컴포넌트가 제공하는 모든 인터페이스에 대해 작성한다.

1.2.1. (인터페이스명)

해당 인터페이스별로 인터페이스 명세를 작성한다.

1.2.1.1. 오퍼레이션

인터페이스가 표준화된 패턴에 따라 작성되었으며, 오퍼레이션의 변화가 없을 경우에는 관련 항목을 생략할 수 있다.

- 오퍼레이션명
오퍼레이션의 명칭을 기술한다.

컴포넌트 구현 설계 작업(2D21)

- 사전조건
해당 오퍼레이션이 설계에 따라 정확히 동작하고 결과를 제공하기 위해 필요한 전제조건 및 가정사항을 기술한다.
- 사후조건
해당 오퍼레이션이 수행된 후 제공되는 결과가 무엇인지를 기술한다.
- 정의 클래스명
해당 오퍼레이션이 정의된 클래스의 명칭을 기재한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 기술시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

2. 기타 클래스

기타 클래스에 대한 명세를 작성한다.

2.1. (클래스명)

해당 클래스별로 클래스 명세를 작성한다. 해당 클래스가 표준화된 구현 매핑 패턴에 따라 작성되었을 경우에는 관련 항목을 생략할 수 있다. 해당 클래스가 재사용 컴포넌트 연계에 따른 클래스이거나 시스템 운용을 위한 기능 추가 등 구현 매핑에 따른 표준화된 패턴을 따르지 않으면 해당 클래스를 중심으로 클래스 다이어그램으로 정적인 구조를 모델링하고 시퀀스 또는 커뮤니케이션 다이어그램으로 동적인 행위를 모델링해 포함한다.

2.1.1. 설명

해당 클래스의 용도, 주요 기능에 초점을 두어 클래스 설명을 기술한다.

2.1.2. 속성

- 속성명
속성의 명칭을 기재한다.
- 가시성
해당 속성의 가시성을 기재한다.
- 타입
해당 속성의 데이터 타입을 기재한다.
- 기본값
객체 생성시 해당 속성의 기본값을 기재한다.

컴포넌트 구현 설계 작업(2D21)

- 설명
속성에 대한 설명을 기술한다. 설명 기술시 '로그인 화면의 비밀번호', '부대부호를 임시로 저장하기 위한 속성' 등 속성이 의미하는 것에 초점을 두어 기술한다.

2.1.3. 오퍼레이션

- 오퍼레이션명
오퍼레이션의 명칭을 기재한다.
- 가시성
해당 오퍼레이션의 가시성을 기재한다.
- 반환 타입
해당 오퍼레이션의 반환 데이터 타입을 기재한다.
- 파라미터 명칭
오퍼레이션 파라미터의 명칭과 데이터 타입을 기재한다.
- 파라미터 설명
오퍼레이션 파라미터의 설명을 기술한다. 파라미터 설명 기술시 해당 파라미터가 의미하는 것에 초점을 두어 기술한다.
- 설명
오퍼레이션에 대한 설명을 기술한다. 설명 기술시 해당 오퍼레이션이 수행하는 또는 처리하는 내역에 초점을 두어 기술한다.

2.1.4. 알고리즘

해당 클래스의 주요 알고리즘을 기술한다.

3. 패키지 정의

구현 클래스를 묶은 패키지를 정의한다.

- 패키지명
구현 클래스 매핑 항목에서 매핑된 구현 클래스의 패키지 명칭을 기재한다.
- 클래스명
해당 패키지에 포함될 구현 클래스의 명칭을 기재한다.
- 설명
해당 패키지에 대한 기타 사항이나 부가적인 내용을 기술한다.

컴포넌트 구현 설계 작업(2D21)

4. 배치 정의

컴포넌트가 실제 시스템을 구성하는 하드웨어에 배치되는 모습을 디플로이먼트 다이어그램을 이용해 모델링하고 배포 단위를 정의한다.

4.1. 디플로이먼트 다이어그램

시스템의 각 하드웨어 구성 요소별로 배포 파일을 할당하여 어떤 컴포넌트가 어떤 하드웨어에 배치되는지를 디플로이먼트 다이어그램을 이용해 작성한다.

4.2. 배포 정의

컴포넌트의 배포 단위를 정의한다. 디플로이먼트 다이어그램에 작성 항목이 포함되어 있을 경우 해당 항목을 생략해도 무방하다.

- 배포 파일명
배포 단위로 묶여지는 물리적인 배포 파일의 명칭을 기재한다.
- 컴포넌트명
배포 파일에 포함되는 컴포넌트 명칭을 기재한다.
- 클래스명
컴포넌트를 구성하는 인터페이스 및 클래스의 명칭을 기재한다.
- 설명
배포 파일과 관련된 기타 사항을 기술한다.

컴포넌트 구현 설계 작업(2D21)

[서식] 트랜잭션 설계서(2D21b)

1. 트랜잭션 목록

컴포넌트 명	인터페이스 명	오퍼레이션 명	트랜잭션 종류	격리수준	예외 처리	
					예외 처리명	Roll-back 여부

컴포넌트 구현 설계 작업(2D21)

■ [항목 설명] 트랜잭션 설계서(2D21b)

1. 트랜잭션 목록

유스케이스별로 명시된 트랜잭션이 어떤 인터페이스 오퍼레이션으로 구현되었는지 기술한다.

- 컴포넌트명
트랜잭션이 정의된 컴포넌트 명칭을 기재한다.
- 인터페이스명
트랜잭션이 제공되는 인터페이스 명칭을 기재한다.
- 오퍼레이션명
트랜잭션이 명시된 오퍼레이션의 명칭을 기재한다. 다중성에 따라 오퍼레이션명이 동일할 경우 파라미터명까지 기재한다.
- 트랜잭션 종류
트랜잭션의 종류를 기재한다.
- 격리수준
트랜잭션의 격리수준을 기재한다.
- 예외 처리명
예외 상황이 발생하였을 때의 에러 코드를 기재한다.
- Roll-back 여부
예외 상황이 발생하였을 때 Roll-back 여부를 기술한다.

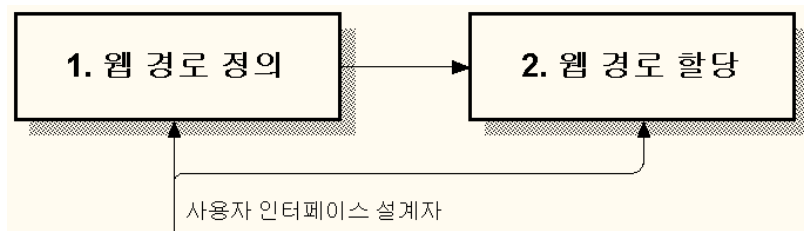
■ 사용자 인터페이스 구현 설계 작업(2D22)

■ 개 요

사용자 인터페이스 설계 결과를 바탕으로 사용자 화면을 플랫폼에 맞게 구현할 수 있도록 웹 페이지 및 사용자 인터페이스 제어 역할을 하는 웹 클래스 등의 경로와 물리적 위치를 정의하고, 웹 클래스 및 웹 페이지를 구성하는 페이지 목록을 설계하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 사용자 인터페이스 명세서
산출물	<ul style="list-style-type: none"> 사용자 인터페이스 설계서
참여자	<ul style="list-style-type: none"> 사용자 인터페이스 설계자 <p>웹 경로를 정의하고 웹 페이지, 웹 클래스 및 웹 컴포넌트를 웹 경로에 할당한다.</p>
관련기법	<ul style="list-style-type: none"> 웹 페이지 모델링

■ 절차 흐름도



■ 절차 설명

1. 웹 경로 정의

소프트웨어 아키텍처 및 컴포넌트 아키텍처를 기반으로 웹 페이지, 클래스 및 컴포넌트가 위치하는 구조를 정의하고 패키징 한다.

- 웹 경로 정의

웹 페이지를 설계하기 위한 이미지, 스크립트, CSS 등이 위치할 디렉토리 구조를 정의하고 웹 클래스가 위치할 논리적인 웹 패키지 구조, 공통 API 혹은 리포팅 도구가 설치된 패키지 구조를 정의한다.

■ 사용자 인터페이스 구현 설계 작업(2D22)

2. 웹 경로 할당

웹 경로에 웹 구성 요소를 할당한다.

- 웹 경로 할당

정의한 웹 패키지 및 경로에 웹 페이지, 웹 클래스 및 웹 컴포넌트를 할당한다.

■ 작업 수행 지침

■ 사용자 인터페이스 구현 설계의 의미

- 사용자 인터페이스 구현 설계는 웹 페이지 등 웹 구성 요소를 패키징하고 경로를 할당해 시스템상의 웹 구성 요소 저장 위치를 정의한다.

■ 사용자 인터페이스 구현 설계시 고려사항

- 웹 페이지를 생성하는 HTML이나 JSP 또는 ASP 파일, 이미지, 사운드와 같은 웹 콘텐츠, 스타일 시트, 스크립트 등 사용자 인터페이스를 구성하기 위해서는 다양한 형태와 목적의 웹 구성 요소가 존재하며 이들 간의 복잡한 호출 관계가 존재한다. 이들의 관리 및 유지보수, 재사용성 등을 고려해 패키징 단위와 경로를 정의해야 한다.

■ 사용자 인터페이스 구현 설계 결과와 이후 작업과의 관계

- 사용자 인터페이스 구현 설계(2D22)를 기반으로 개발자가 구현한 웹 구성 요소를 적정한 위치에 배치한다.

■ 사용자 인터페이스 구현 설계 구성 요소간 관계

- 웹 경로는 하나 이상의 웹 클래스, 웹 페이지, 웹 콘텐츠, 웹 컴포넌트와 관계를 가진다.

■ 사용자 인터페이스 구현 설계 작업(2D22)

■ [서식] 사용자 인터페이스 설계서(2D22a)

1. 웹 경로 목록

타입	경로명	설명

1.1. 웹 클래스 경로 할당

경로명	패키지명	웹 클래스명	파일명

1.2. (웹 구성 요소명) 경로 할당

경로명	(웹 구성 요소명)	파일명

■ 사용자 인터페이스 구현 설계 작업(2D22)

■ [항목 설명] 사용자 인터페이스 설계서(2D22a)

1. 웹 경로 목록

웹 구성 요소를 구분하는 타입별로 웹 경로를 정의하고 할당된 웹 구성 요소를 기술한다.

- 타입
운용 및 유지보수를 고려해 구분한 해당 웹 구성 요소의 패키징 타입을 기재한다.
- 경로명
각 웹 구성 요소가 위치하는 경로를 기재한다.
- 설명
해당 경로에 할당되는 웹 구성 요소에 초점을 두어 경로 설명을 기술한다.

1.1. 웹 클래스 경로 할당

경로별로 할당된 웹 클래스를 기재한다.

- 경로명
각 웹 구성 요소가 위치하는 경로를 기재한다.
- 패키지명
웹 클래스의 패키지 명칭을 기재한다.
- 웹 클래스명
할당된 웹 클래스 명칭을 기재한다.
- 파일명
해당 웹 클래스가 실제로 구현될 파일명을 기재한다.

1.2. (웹 구성 요소명) 경로 할당

※ 웹 페이지, 웹 콘텐츠, 웹 컴포넌트 등 다른 웹 구성 요소는 웹 클래스 경로 할당 서식을 준용해 각 구성 요소의 특성에 맞춰 적용한다.

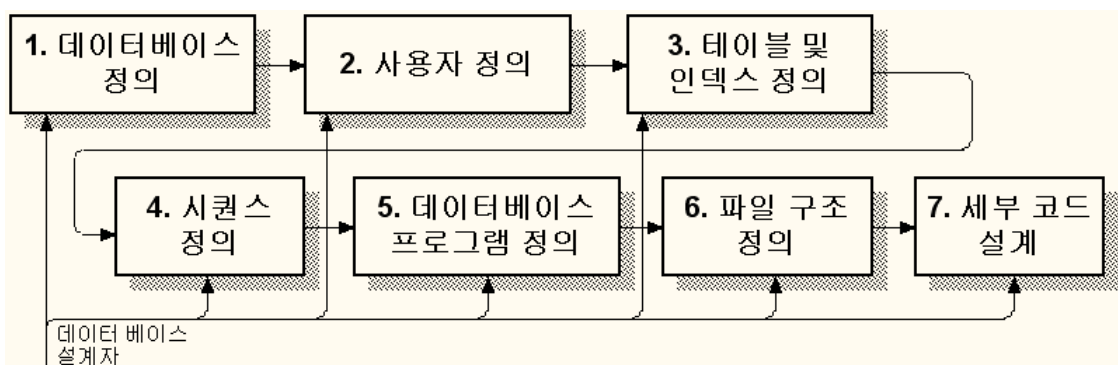
데이터베이스 설계 작업(2D23)

개요

데이터베이스 설계자가 확정된 데이터 모델링 결과를 바탕으로 개발 대상 시스템에서 사용될 DBMS의 환경에 맞는 물리적 스키마와 DBMS 내장 프로그램, 세부 코드를 설계하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 자료 구축 계획서 데이터 명세서 컴포넌트 설계서
산출물	<ul style="list-style-type: none"> 데이터베이스 설계서, (부록)코드 설계서
참여자	<ul style="list-style-type: none"> 데이터베이스 설계자 <p>개발 대상 시스템의 DBMS 환경에 맞는 물리적 데이터베이스 및 데이터베이스 프로그램, 세부 코드를 설계한다.</p>
관련기법	<ul style="list-style-type: none"> 데이터베이스 설계
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 데이터공유환경(SHADE)

절차 흐름도



데이터베이스 설계 작업(2D23)

절차 설명

1. 데이터베이스 정의

데이터베이스관리시스템(DBMS) 특성을 정의한다.

- 데이터베이스 정의
개발 대상 시스템에 적용할 DBMS의 특성과 제공 기능 등 데이터베이스 설계 시 고려해야 할 요소를 정의/식별하고 저장 공간 할당 등의 물리적 특성을 정의한다. 공간은 최초 구분한 후 통합, 분리 등의 변경이 쉽지 않으므로 면밀한 분석을 실시하여 구분한다.

2. 사용자 정의

데이터베이스의 사용자 계정을 정의한다.

- 데이터베이스 사용자 정의
비즈니스 관점이 아닌, 시스템 관점으로 데이터베이스 트랜잭션을 유발하는 사용자(어플리케이션 포함)를 일반화하여 최소한의 계정을 추출해낸다.

3. 테이블 및 인덱스 정의

데이터 모델링 결과를 특정 DBMS에 맞는 테이블 또는 객체 형태로 변환하여 정의한다.

- 물리 데이터 모델 설계
논리 데이터 모델의 엔티티명을 테이블로 식별하고 논리 데이터 모델의 속성을 컬럼으로 식별해 컬럼의 데이터 타입, 길이, 제약 조건 등을 DBMS에 맞도록 전환하여 초기 물리 데이터 모델을 정의한다.
초기 물리 데이터 모델을 기반으로 처리량, 데이터베이스의 성능, 컬럼의 제약 조건 등을 고려해 테이블을 정제하여 물리 데이터 모델을 설계한다.
- 데이터베이스 매핑
물리 데이터모델의 테이블별로 데이터베이스 상에서 테이블을 관리하기 위한 ID, 소유권과 타 사용자의 권한을 정의하고 테이블이 위치하는 공간을 지정한다.
- 인덱스 정의
테이블의 컬럼 중에서 인덱스를 생성할 컬럼을 선정하여 인덱스 컬럼의 값들과 그 값들을 포함하고 있는 레코드의 논리적 주소를 연관시켜 만들어지는 인덱스에 대한 세부사항을 정의한다.

데이터베이스 설계 작업(2D23)

4. 시퀀스 정의

필요시 시퀀스를 정의한다. 시퀀스는 일련번호를 생성해낼 수 있는 객체로써 DBMS에 따라 시퀀스의 개념을 지원하지 않을 경우는 해당 사항이 없다.

- 시퀀스 정의

시퀀스의 시작값, 증가값 등 특성을 정의하고 소유권과 타 사용자의 권한을 정의한다.

5. 데이터베이스 프로그램 정의

뷰, 패키지, 프로시저 등 데이터베이스 프로그램을 정의한다.

- 뷰 정의

불필요한 중복 테이블 정의 방지, 쿼리 사용 빈도와 성능 요구에 따라 뷰를 정의한다. 뷰는 테이블 또는 다른 뷰에 대한 조회 조건을 사전에 정의하여 메모리에 상주시킴으로 디스크 기반 DBMS에서 신속한 조회를 보장할 수 있다.

- 패키지 정의

객체 의존성을 제거하기 위해 패키지를 정의한다. 패키지는 프로시저 또는 함수를 묶어주는 역할을 한다. 만일 개발 대상 시스템 DBMS가 패키지를 지원하지 않거나 특성이 아래와 다르다면 해당 사항이 없다.

- 프로시저 정의

데이터베이스 층에서 필요한 프로그램 로직을 위해 프로시저를 정의한다. 프로시저는 사전에 정의되어 데이터베이스 층에서 수행되는 절차적 프로그래밍 언어이다.

굳이 어플리케이션에서 실행될 필요가 없는 로직(예를 들어 데이터 합산 등)을 데이터베이스 층에서 실행하면 불필요한 입출력을 제거할 수 있으므로 처리 속도가 향상되고 로직이 집중화하며 일관성을 확보할 수 있다.

만일 개발 대상 시스템 DBMS가 프로시저를 지원하지 않으면 해당 사항이 없다.

- 함수 정의

데이터베이스 층에서 필요한 프로그램 로직을 위해 함수를 정의한다. 함수는 프로시저와 마찬가지로 데이터베이스 층에서 수행되는 절차적 프로그래밍 언어로써, 프로시저는 입력 변수만 존재하지만 함수는 입력 외에도 출력 변수를 가지고 있는 점이다. (DBMS에 따라 다소 차이가 있음.)

데이터베이스 설계 작업(2D23)

- 트리거 정의
자동으로 데이터의 변화를 감지하기 위해 트리거를 정의한다. 트리거는 지정된 테이블의 데이터가 변화될 때마다 다른 데이터 처리가 동시에 일어날 수 있도록 지원하는 데이터베이스 프로그램이다. 만일 개발 대상 시스템 DBMS가 트리거를 지원하지 않으면 해당 사항이 없다.
- 스케줄 정의
일정 시간별로 실행되는 로직을 위해 스케줄을 정의한다. 스케줄은 DBMS에서 일정 시간을 기반으로 실행될 수 있는 프로그램으로써 데이터를 조작하고 다른 프로시저나 함수를 호출할 수 있다.
DBMS의 스케줄은 운영체제의 스케줄과 비슷한 특성을 가지고 있으므로 개발 대상 시스템 특성에 따라 운영체제 또는 DBMS의 스케줄을 선택하여 정의한다.

6. 파일 구조 정의

파일 시스템을 사용하는 경우 파일 구조 및 디렉토리 위치 등을 정의한다.

- 파일 구조 정의
파일 시스템을 이용해 저장할 자료의 항목을 확인하고 해당 항목을 저장, 수정, 삭제 및 조회하기 위한 규칙을 정의한다. 또한 파일이 위치할 디렉토리를 정의한다.

7. 세부 코드 정의

개발 대상 시스템에서 사용할 코드에 대해 세부 코드 값을 정의한다.

- 세부 코드 정의
논리 및 물리 데이터 모델링 활동을 통해 식별된 코드를 대상으로 값을 구성하는 세부 코드를 정의한다.
국방 표준 코드는 ‘국방상호운용성관리 지시’에 의거하여 국방메타데이터관리체계에서 제시하고 있는 ‘데이터 표준화 지침서’에 따라 코드 정의 절차를 수행한다.

■ 작업 수행 지침

■ 사용자 계정 설계시 고려사항

- 일반적으로 사용자가 접근하는 모든 객체는 권한에 대한 체크를 하므로 사용자 계정을 시스템 생성 공간이 아닌, 별도 공간에 생성하는 것이 수행 속도상 유리하다.

데이터베이스 설계 작업(2D23)

■ 테이블 및 인덱스 정의시 고려사항

- 개발 대상 시스템이 연동 소요를 가진다면 응용 테이블뿐만 아니라 연동 관련 메타정보 테이블(예를 들어 에러 로그, 코드 정의, 기관 식별자, 연동체계 식별자 등)에 대한 정의도 함께 수행한다.
- 테이블 및 컬럼 등에 대한 명명규칙은 '국방상호운용성관리 지시'에 의거하여 국방메타데이터관리체계에서 제시하고 있는 '데이터 표준화 지침서'의 명명규칙을 준수해야 한다.

■ 컬럼 제약조건 도출시 고려사항

- 데이터의 정합성과 정확성을 높이기 위해서는 어플리케이션의 UI층, 비즈니스 컨트롤 층, 데이터베이스 층에서 각각 제어를 해야 한다. 이에 따라 데이터베이스에서 각 컬럼의 제약조건을 충분히 도출해서 어플리케이션의 제약조건과 맞춰야 한다.
- 이를 생략하고 긴급하게 설계를 진행하는 경우가 빈번하다. 이렇게 되면 데이터의 품질이 떨어지고 뒤늦게 제약조건을 추가하면 어플리케이션에 제약조건이 반영되지 않아 실행 오류를 일으킬 가능성이 있다.
- 각 컬럼의 Not Null 조건, 유효값, 기본값, 코드화 등의 제약조건을 충분히 검토하여 도출해야 한다.

■ 인덱스 설계시 유의사항

- 데이터의 입력, 수정, 삭제가 자주 발생하는 컬럼을 인덱스(Index)로 지정하면 데이터 영역뿐만 아니라 인덱스에 있는 정보도 DBMS에서 수정해야 하므로 데이터 영역에 대해서만 프로세싱이 발생할 때보다 시간과 자원이 더 소모된다.
- 또한 하나의 테이블에 너무 많은 인덱스가 정의되어 있으면 데이터를 조회하여 처리하기에는 좋은 성능을 가지지만 데이터의 수정이 발생한 경우에는 오히려 성능이 저하될 수 있고 디스크 용량도 많이 차지하게 된다.
- 복합 인덱스(Composite Index)를 생성할 때는 컬럼의 순서에 따라서 성능에 많은 영향을 미친다. 복합 인덱스의 앞쪽에 조회 범위를 줄일 수 있는 컬럼이 위치하도록 설계해야 한다.
- 앞쪽에 위치할 수 있는 컬럼은 항상 사용되어야 하며, 가능하면 '='로 비교되도록 하고 분포도가 좋은 컬럼이 되도록 선정해야 한다. A, B, C 컬럼이 복합 인덱스로 선정되었지만 SQL문장의 WHERE절에서 B와 C에 상수값이 들어오고, A 값은 들어오지 않는다면 인덱스는 사용되지 않는다. 그러나 A와 B에 값이 들어오고, C에 값이 들어오지 않는다면 인덱스는 사용될 수 있다.

데이터베이스 설계 작업(2D23)

■ 뷰 활용시 고려사항

- 뷰의 남용은 문제를 야기시킬 수 있다. 예를 들면 유지 관리할 뷰의 중복, 뷰나 컬럼 이름의 혼동 및 특정 뷰와 관련된 기능상의 제약(예: 뷰를 통한 갱신) 등이 있을 수 있으므로 이러한 점을 고려하여 뷰를 정의한다.
- 너무 많은 뷰를 사용하는 것은 메모리를 점유함으로써 메모리 할당 및 활용 측면에서 부정적인 영향이 있을 수 있다.

■ 패키지 정의시 고려사항

- 프로시저나 함수가 특정 객체(테이블 등)에 대한 내용이 포함되어 있을 경우, 이 객체의 형상이 변화되었다면 프로시저와 함수는 불능 상태에 빠지게 된다.
- 프로시저와 함수가 패키지 내에 포함되는 형태라면 패키지 자체는 객체의 형상이 변경되었더라도 여전히 사용이 가능하고 패키지 바디에 해당하는 프로시저와 함수만 불능 상태가 된다.
- 패키지 없이 선언된 프로시저와 함수를 이용하는 쿼리 문장은 하드 파싱을 수행하기 때문에 객체의 형상 변경시 재컴파일되고 자원 손실을 유발하여 시스템의 장애를 초래한다. 따라서 객체에 대한 의존성을 제거하기 위해 가급적 패키지를 사용하여 장애 상황을 예방해야 한다.

■ 함수 정의시 고려사항

- 코드값과 코드명을 변환하기 위한 함수 등 빈번하게 사용되는 로직을 데이터베이스 층에서 구현하면 쿼리에서 바로 사용할 수 있으므로 불필요한 입출력과 데이터베이스 접속 수요를 줄일 수 있고 프로그램 코드가 간결해진다.

■ 트리거 활용시 고려사항

- 트리거는 테이블 데이터의 변화 여부를 DBMS에서 판단하므로 로직이 집중되고 일관성을 확보할 수 있으며, 데이터베이스 층에서 작성되므로 상대적으로 간편하게 로직을 구현할 수 있다.
- 트리거는 DBMS에 따라 정확성을 확보하기 위한 조건이 까다로울 수 있다. 예를 들어 관련된 테이블의 형상이 계속 변화하거나 트랜잭션에 과부하가 걸리면 불능 상태가 되거나 테이블을 잠금(Lock) 상태로 만들 가능성이 높다.
- 위와 같은 상황이 발생하였을 경우 어플리케이션에 에러 메시지를 제공하지 않을 수 있으므로 사용자는 문제가 생겼다는 것을 인지하지 못하고 이로 인해 오히려 데이터의 정합성이 떨어지는 원인이 될 수 있다. 따라서 상대적으로 중요하지 않은 로그 기록 등 제한적인 목적으로 활용되어야 한다.

데이터베이스 설계 작업(2D23)

■ 세부 코드 정의시 고려사항

- 코드는 다양하게 나타날 수 있는 데이터 값을 정형화하기 위해, 정의된 기준에 따라 제한된 범위 내의 기호로 대치한 것을 의미하며, 시스템 내의 코드는 해당 업무관련 내용을 일정기준에 의하여 간소하게 처리할 수 있도록 단순화하고, 유일성을 보장하기 위해 코드값과 코드값 의미를 1:1로 정하여 체계 내에서 활용되고 있는 데이터를 의미한다.
- 코드는 국방 표준 코드와 각 시스템별 코드가 존재한다. 데이터베이스 설계자는 국방 표준 코드를 검색하여 적용성을 판단하고 적정한 코드가 없을시 각 시스템 고유의 코드를 설계해야 한다. 코드 설계시에는 다음 사항에 유의한다.
 - 최소의 자릿수
 - 고유성
 - 분류의 편리성
 - 일관성
 - 기계 처리의 용이성
 - 표의성
 - 확장 용이성

■ 초기 데이터베이스 용량 산정

- 데이터베이스 공간 할당을 위해 초기 데이터베이스의 용량을 산정하여 데이터베이스 설계서에 기재한다. 데이터베이스 용량 산정법은 시스템의 특성과 DBMS에 따라 가변적이므로 DBMS 기술 문서 등 별도 내용을 참고하여 실시한다. 부가적으로 도입되는 하드웨어의 용량이 데이터베이스 구축에 적정한지를 판단한다.
- DBMS가 공간별 자동 용량 증가(Auto Extent) 기능을 지원하면 개별적 테이블에 대한 초기 용량(또는 공간 크기) 산정은 실시하지 않는다. 만일 자동 용량 증가 기능이 없고 시스템 특성상 초기 및 향후 증가 용량에 대한 예측이 중요할 경우에는 해당 작업을 실시하고 산정 과정 및 결과에 대한 산출물을 추가로 작성한다.
- 용량 산정시에는 대체로 일방적으로 증가하는 정보군(거래, 계약 정보 등), 빈번히 수정되는 정보군(개인 이력, 상품 특성 등), 초기 설정 후 변경과 증가가 좀처럼 발생하지 않는 정보군(우편번호, 직급체계 등), DB 인덱스 설정용 공간, Log 처리용 등의 정보군으로 분류한 뒤 해당 DBMS의 용량 산정 방법에 따라 산정한다.

■ 데이터베이스 설계 작업(2D23)

■ 데이터베이스 암호화 관련 산출물 작성

- 만일 데이터베이스가 상용 소프트웨어 또는 자체적 로직에 의해 암호화될 경우 암호화 방식, 암호화 대상 등에 대한 내용을 별도의 산출물로 작성하여 데이터베이스 설계서의 부록으로 첨부한다.

■ 추가 내용의 산출물 작성

- DBMS는 제조사와 제품, 버전에 따라 정의 및 기술되어야 하는 내용이 다양하고 일부 기능을 지원하지 않는 제품이 존재하므로 각 사업의 특성에 맞게 산출물에 해당 내용을 추가로 기술해야 한다. (예 : synonym, 사용자 role 등)

■ 운전자 지침서 초안 작성 시기 조정

- 데이터베이스의 데이터 백업 및 복구 방법이 설계 단계에서 정밀하게 도출되어야 한다면 별도 산출물을 작성하거나 3T52a 운전자 지침서의 일부 초안을 설계 단계에서 작성할 수 있다.

■ 모델링 파일 제출

- 일반적으로 모델링 도구를 이용하여 데이터베이스를 설계하므로 세부 정보 확인, 형상 관리 및 원활한 유지보수 활동을 위해 산출물로 데이터베이스 설계서외에도 모델링 파일을 함께 제출해야 한다.
- ‘국방상호운용성관리 지시’에 따라 확정된 모델링 파일은 국방메타데이터관리체계에 등록 요청한다.

데이터베이스 설계 작업(2D23)

[서식] 데이터베이스 설계서(2D23a)

1. 데이터베이스

1.1. DBMS

DBMS명	설 명	특수 요건

1.2. 데이터베이스 공간

DBMS명	공간명	설 명	파일 경로	공간 크기

1.3. 사용자

사용자명	설 명	사용기한	데이터베이스 공간명

2. 물리 데이터 모델

2.1. 물리 데이터 모형

2.2. 테이블 및 컬럼

2.2.1. (테이블ID)

테이블 ID		테이블명		공간명	
설 명					
소유권자			사용자 및 권한		
컬럼명	설명	키 유형	Not Null	데이터 타입 및 길이	제약조건

데이터베이스 설계 작업(2D23)

2.3. 인덱스

2.3.1. (인덱스ID)

인덱스ID	인덱스 형태	공간명
테이블ID	컬럼명(우선순위)	ASC/DESC
설 명		

2.4. 시퀀스

시퀀스ID	시작값	증가값	최대값	최소값	반복 여부	소유권자	사용자 및 권한

3. 데이터베이스 프로그램

3.1. 뷰

3.1.1. 뷰 구조

3.1.2. (뷰ID)

뷰ID	뷰명	수정가능 여부	
설명			
소유권자		사용자 및 권한	
컬럼명	설명	유도 테이블/뷰ID	관계식 및 제약조건

데이터베이스 설계 작업(2D23)

3.2. 패키지

3.2.1. (패키지ID)

패키지ID	설 명	소유권자	사용자 및 권한
구성 요소	구성 요소 구분	구성 요소ID	

3.3. 프로시저

3.3.1. (프로시저ID)

프로시저ID	예외처리	소유권자	사용자 및 권한
설 명			
파라미터	데이터	타입 및 길이	입출력 유형

3.4. 함수

3.4.1. (함수ID)

함수ID	예외처리	소유권자	사용자 및 권한
설 명			
파라미터		리턴값	
데이터	타입 및 길이	데이터	타입 및 길이

데이터베이스 설계 작업(2D23)

3.5. 트리거

3.5.1. (트리거ID)

트리거ID	실행 시점	실행 조건	관련 테이블	실행 범위	소유권자
설 명					
예외처리					

3.6. 스케줄

스케줄ID	설 명	실행 시간

4. 파일 구조

하드웨어명	파일 경로	파일 경로 설명	파일 타입	파일 설명

데이터베이스 설계 작업(2D23)

■ [항목 설명] 데이터베이스 설계서(2D23a)

1. 데이터베이스

1.1. DBMS

- DBMS명
DBMS를 대표하는 명칭 및 버전을 기재한다.
- 설명
제조사를 포함하여 DBMS에 대한 설명을 기술한다.
- 특수 요건
가용성, 보안, 운용방식 등에 대해서 기술한다.

1.2. 데이터베이스 공간

- DBMS명
DBMS를 대표하는 명칭을 기재한다.
- 공간명
데이터베이스 공간을 대표하는 명칭을 기재하되, DBMS가 자체 관리를 위해 기본으로 생성하는 공간은 제외한다.
- 설명
데이터베이스 공간에 대한 설명을 기술한다.
- 파일 경로
데이터베이스 공간 또는 이를 구성하는 파일이 위치하는 파일 경로를 기재한다.
- 공간 크기
데이터베이스 공간의 크기를 기재한다.

1.3. 사용자

- 사용자명
관리자를 포함하는 DBMS 사용자 계정명을 기재하되, DBMS가 기본으로 생성하면서 개발에서 불필요한 계정은 제외한다.
- 설명
사용자에 대한 설명을 기술한다.
- 사용기한
사용자의 권한이 기한을 가졌을 경우 기한을 명시한다.
- 데이터베이스 공간명
사용자 계정이 생성된 공간명을 기재한다.

데이터베이스 설계 작업(2D23)

2. 물리 데이터 모델

2.1. 물리 데이터 모형

물리 ER 다이어그램으로 작성하되, 테이블, 컬럼, DBMS에 맞는 데이터 타입 및 길이, 도메인명이 식별될 수 있도록 작성하고 DBMS가 여러 개일 경우 이에 대한 구분 표시를 한다.

2.2. 테이블 및 컬럼

각 테이블에 대하여 테이블과 컬럼에 대한 특성들을 정의한다. 데이터 설계서에 서 데이터 모형의 엔티티 및 속성을 기반으로 작성하고 DBMS가 여러 개일 경우 이에 대한 구분 표시를 한다.

2.2.1. (테이블ID)

각 테이블별로 작성한다.

- 테이블ID
테이블을 식별할 수 있는 식별자를 기재한다.
- 테이블명
논리 엔티티에 대응되는 테이블의 대표 명칭을 기재한다.
- 공간명
테이블이 생성된 데이터베이스 공간명을 기재한다. 여러 공간에 나뉘어 생성된 테이블의 경우, 해당하는 공간명을 모두 기재한다.
- 설명
테이블에 대한 간단한 설명을 기술하되, 시스템 사용자, 데이터 보관 시간 단위 등을 포함한다.
- 소유권자
테이블을 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 테이블을 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.
- 컬럼명
논리 엔티티 속성에 대응되는 컬럼의 명칭을 기재한다.
- 설명
컬럼에 대한 간단한 설명 또는 논리 엔티티 속성명을 기재한다.

데이터베이스 설계 작업(2D23)

- 키 유형
컬럼의 키 여부와 유형으로써 주키(PK : Primary Key), 외래키(FK : Foreign Key)를 기재하고 주키일 경우 주키 순서 번호를 함께 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.
- 데이터 타입 및 길이
DBMS에 맞는 컬럼의 데이터 타입 및 길이를 기재한다.
- 제약조건
필요 시 각 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.

2.3. 인덱스

테이블의 컬럼값과 그 값을 포함하고 있는 레코드의 논리적 주소를 연관시켜 만들어지는 인덱스에 대한 세부사항을 정의한다. DBMS가 여러 개일 경우 이에 대한 구분 표시를 한다.

2.3.1. (인덱스ID)

- 인덱스ID
인덱스를 식별할 수 있는 식별자를 기재한다.
- 인덱스 형태
인덱스가 유일성 인덱스(unique index)인지 비유일성 인덱스(nonunique index)인지와 클러스터된 인덱스 여부를 기재한다.
- 공간명
인덱스가 생성된 데이터베이스 공간명을 기재한다. 여러 공간에 나뉘어 생성된 인덱스의 경우, 해당하는 공간명을 모두 기재한다.
- 테이블ID
인덱스를 사용하는 테이블ID를 기재한다.
- 컬럼명
인덱스를 구성하는 컬럼명을 모두 기재하되, 우선 순서대로 기재한다.
- ASC/DESC
인덱스가 순서 인덱스(ordered index)인 경우 인덱스 테이블의 컬럼 값이 오름차순이면 'ASC'를 기입하고 내림차순이면 'DESC'를 기재한다.
- 설명
인덱스의 기능에 대하여 간단한 설명을 기술한다.

데이터베이스 설계 작업(2D23)

2.4. 시퀀스

시퀀스는 일관된 연속값을 가질 수 있는 객체이다. 필요에 따라 시퀀스를 생성할 경우, 이에 대한 속성을 정의한다. 시퀀스를 지원하지 않는 DBMS는 해당 사항이 없다.

- 시퀀스ID
시퀀스를 식별할 수 있는 식별자를 기재한다.
- 시작값
시퀀스가 생성될 때 시작되는 값을 기재한다.
- 증가값
시퀀스가 증가될 때 증가되는 값을 기재한다.
- 최대값
시퀀스가 최대로 가질 수 있는 값을 기재한다.
- 최소값
시퀀스가 최소로 가질 수 있는 값을 기재한다.
- 반복여부
시퀀스가 최대값 또는 최소값에서 증가될 때 시작값으로 반복될 수 있는지 여부를 기재한다.
- 소유권자
시퀀스를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 시퀀스를 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.

3. 데이터베이스 프로그램

DBMS가 지원하는 내장 프로그램으로써 이를 이용할 경우 구현 이전에 이에 대한 기본 설계 또는 명세를 작성한다. DBMS가 여러 개일 경우 이에 대한 구분 표시를 한다.

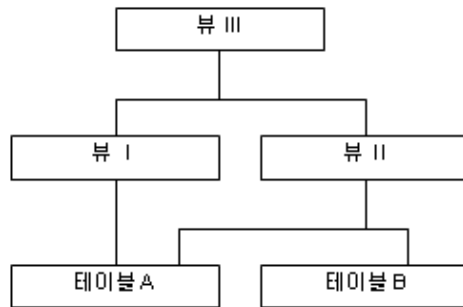
3.1. 뷰

특정한 목적의 쿼리를 뷰로 정의할 경우 이의 구조와 유도 관계를 정의한다. 뷰 속성에서 관계식 및 제약조건이 복잡해서 아래 서식으로 표현이 불가능할 경우 다른 방법(예 : 뷰 생성 쿼리 문장) 으로 작성할 수 있다.

데이터베이스 설계 작업(2D23)

3.1.1. 뷰 구조

뷰를 구성하는 테이블이나 또 다른 뷰와의 관계를 계층 구조 형식으로 나타낸다.



<계층 구조로 작성된 예>

3.1.2. (뷰ID)

- 뷰ID
뷰를 식별할 수 있는 식별자를 기재한다.
- 뷰명
뷰를 대표할 수 있는 명칭을 기재한다.
- 수정 가능 여부
뷰에 의한 데이터 수정 가능 여부를 기재한다. 뷰의 수정 가능 조건은 각 DBMS 매뉴얼을 참고한다.
- 설명
뷰에 대한 간략한 설명을 기술하되, 뷰의 사용 목적을 포함한다.
- 소유권자
뷰를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 뷰를 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.
- 컬럼명
뷰를 구성하는 컬럼명을 기재한다.
- 설명
각 뷰 컬럼에 대한 간략한 설명을 기술한다.
- 유도 테이블/뷰ID
각 뷰 컬럼이 유도되는 테이블이나 뷰의 ID를 기재한다.

데이터베이스 설계 작업(2D23)

- 관계식 및 제약조건
원천 테이블/뷰의 컬럼으로부터 현재 뷰의 컬럼으로 유도되기 위한 관계식(연산 등) 또는 제약조건(예 : WHERE 조건절)등에 대해 기술한다.

3.2. 패키지

패키지는 DBMS에 따라 지원하지 않을 수 있거나 다른 형태로 정의되므로 DBMS에 따라 적절히 서식을 변형하여 기술한다.

3.2.1. (패키지ID)

- 패키지ID
패키지를 식별할 수 있는 식별자를 기재한다.
- 설명
패키지에 대한 간략한 설명을 기술한다.
- 소유권자
패키지를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 패키지를 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.
- 구성 요소 구분
패키지를 구성하는 요소에 대한 구분을 기재한다.(예 : 함수, 프로시저 등)
- 구성 요소ID
구성 요소를 대표하는 ID를 기재한다.

3.3. 프로시저

패키지에 속한 프로시저와 속하지 않은 프로시저를 모두 기술한다.

3.3.1. (프로시저ID)

- 프로시저ID
프로시저를 식별할 수 있는 식별자를 기재한다.
- 예외처리
프로시저 수행에서 예외가 발생했을 경우 처리 방법을 기술한다.

데이터베이스 설계 작업(2D23)

- 소유권자
프로시저를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 프로시저를 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.
- 설명
프로시저가 수행하는 기능에 대한 상세한 설명을 기술한다.
- 파라미터-데이터
데이터의 논리명을 기재한다.
- 파라미터-타입 및 길이
데이터의 타입 및 길이를 기재한다.
- 파라미터-입출력 유형
파라미터가 입력/출력 또는 입출력 용도로 사용되는지를 기재한다.

3.4. 함수

패키지에 속한 함수와 속하지 않은 함수를 모두 기술한다.

3.4.1. (함수ID)

- 함수ID
함수를 식별할 수 있는 식별자를 기재한다.
- 예외처리
함수 수행에서 예외가 발생했을 경우 처리 방법을 기술한다.
- 소유권자
함수를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 사용자 및 권한
소유권자 외에 함수를 사용할 수 있는 사용자와 그에 대한 권한을 각각 기재하되, 권한은 차등적 또는 부분적인 권한을 가졌을 경우에만 기재한다.
- 설명
함수가 수행하는 기능에 대한 상세한 설명을 기술한다.
- 파라미터-데이터
데이터의 논리명을 기재한다.
- 파라미터-타입 및 길이
데이터의 타입 및 길이를 기재한다.

데이터베이스 설계 작업(2D23)

- 리턴값-데이터
데이터의 논리명을 기재한다.
- 리턴값-타입 및 길이
데이터의 타입 및 길이를 기재한다.

3.5. 트리거

테이블의 자료가 변화할 때 이에 따라 자동으로 데이터 조작 및 다른 프로시저나 함수 등을 호출 할 수 있는 프로그램이다.

3.5.1. (트리거ID)

- 트리거ID
트리거를 식별할 수 있는 식별자를 기재한다.
- 실행 시점
테이블에 자료 변화가 일어날 때 트리거가 실행되는 시점(변화 전, 후)을 기재한다.
- 실행 조건
트리거가 실행되는 자료 변화 조건(삽입, 갱신, 삭제)을 기재한다.
- 관련 테이블
트리거가 실행되기 위한 조건으로써 자료가 변화되는 테이블의 명칭을 기재한다.
- 실행 범위
하나의 DML문에 의해 트리거가 적용되는 범위(문장, 행)를 기재한다. 문장 트리거는 하나의 DML에서 단 한번만 실행되며 컬럼의 각 데이터 행을 제어할 수 없다. 행 트리거는 각 데이터 행에 변화가 생길 때마다 실행되며 그 데이터 행의 실제 값을 제어할 수 있다.
- 소유권자
트리거를 생성한 소유권자를 기재하되, 데이터베이스 사용자명으로 기재한다.
- 설명
트리거가 수행하는 기능에 대한 상세한 설명을 기술한다.
- 예외처리
트리거 수행에서 예외가 발생했을 경우 처리 방법을 기술한다.

3.6. 스케줄

스케줄은 DBMS에 따라 지원하지 않을 수 있거나 다른 형태로 정의되므로 DBMS에 따라 적절히 서식을 변형하여 기술한다.

데이터베이스 설계 작업(2D23)

- 스케줄ID
스케줄을 식별할 수 있는 식별자를 기재한다.
- 설명
스케줄이 수행하는 기능에 대한 상세한 설명을 기술한다.
- 실행 시간
스케줄이 실행되는 시간(매 시 또는 시간 반복 등)에 대해 기재한다.

4. 파일 구조

파일 시스템을 사용하는 경우 파일 구조 및 디렉토리 위치 등을 기술한다.

- 하드웨어명
파일 시스템이 위치하게 되는 하드웨어명을 기재한다.
- 파일 경로
하드웨어에서 파일 시스템이 위치하게 되는 디렉토리 경로를 기재한다.
- 파일 경로 설명
파일 경로에 위치한 파일 시스템에 대한 간략한 설명을 기술한다.
- 파일 타입
디렉토리내 각 파일 타입(예 : XML, CSV 등)을 기재한다.
- 파일 설명
각 파일 타입의 사용 목적을 기술하되, 게시판 프로그램 등에서 첨부 파일에 대한 파일 타입 및 설명은 생략한다.

데이터베이스 설계 작업(2D23)

[서식] 부록 - 코드 설계서

1. 코드 목록

코드ID	코드명	코드 영문명	코드 영문약어명	설 명	데이터타입	데이터길이	표준 코드 여부	비고

2. 세부 코드

코드ID	코드명	코드값	코드값명	상위 코드명	상위 코드값	상위 코드값명

데이터베이스 설계 작업(2D23)

■ [항목 설명] 코드 설계서

1. 코드 목록**

코드 표준화 작업에 대한 세부적인 지침은 ‘국방상호운용성관리 지시’에 의거하여 국방메타데이터관리체계에서 제시하고 있는 ‘데이터 표준화 지침서’를 참조한다.

- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재한다.
- 코드명
코드를 대표하는 명칭을 기재한다.
- 코드 영문명
코드의 영문명을 기재한다.
- 코드 영문약어명
코드의 영문명에 대한 약어명을 기재한다.
- 설명
코드에 대한 개략적인 설명을 기술한다.
- 데이터 타입
코드의 논리적 데이터 타입(문자형, 숫자형, 날짜형 등)을 나타내며 각 DBMS의 특성에 맞는 타입을 지정한다.
- 데이터 길이
코드의 최대 데이터 길이를 기재한다.
- 표준 코드 여부
해당 코드가 국방메타데이터관리시스템(MDR)의 표준 코드일 경우 ‘Y’를 기술하고 그렇지 않을 경우 ‘N’을 기재한다.
- 비고
그 밖의 또 다른 항목이 존재하는 경우 해당 내용을 기재한다.

2. 세부 코드**

세부 코드에 대한 정의를 기술한다. ‘국방상호운용성관리 지시’에 의거하여 국방메타데이터관리시스템(MDR)에서 관리되는 표준 코드일 경우, 세부 코드 정의를 생략할 수 있다.

- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재한다.

** 데이터 표준화 지침 유관 항목

데이터베이스 설계 작업(2D23)

- 코드명
코드를 대표하는 명칭을 기재한다.
- 코드값
코드를 구성하는 세부 코드의 실제 값을 기재한다.
- 코드값명
코드값이 의미하는 명칭을 기재한다.
- 상위 코드명
해당 코드가 계층코드인 경우, 상위분류에 해당되는 코드명을 기재한다.
- 상위 코드값
해당 코드가 계층코드인 경우, 상위분류에 해당되는 코드의 실제 값을 기재한다.
- 상위 코드값명
상위 코드값이 의미하는 명칭을 기재한다.

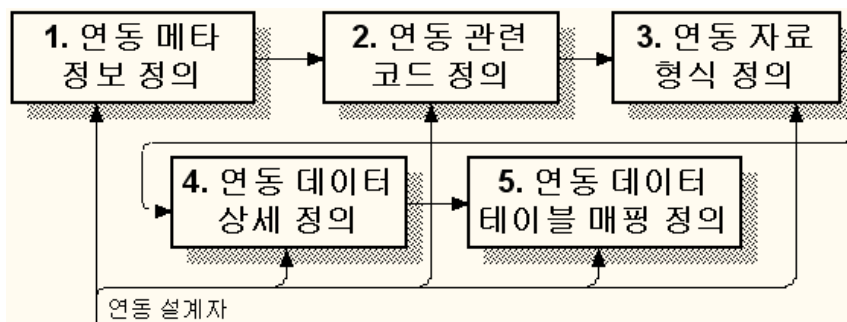
연동 설계 작업(2D24)

개 요

시스템간 연동을 구현하기 위해 연동 항목의 물리적 설계와 연동 시스템 또는 모듈의 아키텍처 구성 요소를 구체화하고 공통 표준 및 각 매핑 규칙을 확정하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 연동 소요 정의서 컴포넌트 설계서 데이터베이스 설계서, (부록)코드 설계서
산출물	<ul style="list-style-type: none"> 연동 설계서, (부록)연동 항목 테이블 매핑 설계서
참여자	<ul style="list-style-type: none"> 연동 설계자 <p>연동 항목에 대한 물리적 설계와 공통 표준 및 매핑 규칙을 설계한다.</p>
관련기법	해당 사항 없음
상호운용성 관련사항	<ul style="list-style-type: none"> 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 연동 메타 정보 정의

연동 테이블 목록, 연동 상황, 연동 로그 등 연동에 대한 메타 정보를 기록하는 테이블을 정의한다.

연동 설계 작업(2D24)

- 연동 메타 정보 정의

메타 정보로 관리할 항목을 식별하고 이를 저장/관리하기 위한 테이블을 정의한다. 연동에 관한 정보만을 구분하여 식별성을 높이기 위해 산출물을 작성할 권장하지만 만일 데이터베이스 설계서와 통합이 필요하다면 연동 설계서에서는 생략될 수 있다

2. 연동 관련 코드 정의

연동 과정, 오류 등을 식별할 수 있는 코드를 정의하고 시스템간 코드 자료에 대한 매핑을 실시한다.

- 연동 관련 코드 정의

데이터베이스 설계와 마찬가지로 연동 시스템에서 공통적으로 사용되는 코드를 정의한다. 연동에 대한 코드만을 구분하여 식별성을 높이기 위해 산출물을 작성할 권장하지만 만일 데이터베이스 설계서의 부록인 코드 설계서와 통합이 필요하다면 연동 설계서에서는 생략될 수 있다.

- 오류 메시지 정의

연동 과정에서 발생할 수 있는 오류를 식별하고 이를 코드로 정의한다.

- 코드 자료 매핑 설계

연동 대상 시스템에서 사용하는 코드가 개발 대상 시스템의 코드와 불일치할 경우 이를 매핑하기 위한 코드 매핑을 정의한다.

3. 연동 자료 형식 정의

연동 자료의 공통 내용이나 메시지 형식과 같이 표준 형식에 대한 정의를 실시한다.

- 연동 테이블 공통 컬럼 정의

연동 항목별로 테이블을 정의할 경우, 이에 공통적으로 들어가는 컬럼(예 : EAI 일련번호, 전송일시, 전송 구분자 등)을 정의한다.

- 연동 메시지 형식 정의

연동 메시지 형식 정의는 전문 메시지나 SAM(Sequential Access Method, 순차접근방법) 파일과 같이 정보가 특정 순서로 전송되어야 하는 경우(예 : 헤더에 암호화 키, 메시지 식별자, 버전 정보 등을 포함), 또는 전자 문서 교환 시스템의 경우 DTD(Document Type Definition)와 같이 데이터 구성 요소가 전체적으로 특정 형식을 준수해야 하는 경우, 이에 대한 형식을 정의한다.

연동 설계 작업(2D24)

4. 연동 데이터 상세 정의

연동되는 인터페이스 데이터에 대한 상세 규약을 정의한다.

- 연동 데이터 상세 정의
분석 단계에서 작성된 연동 항목 소요서(IER)의 각 연동 항목에 대한 물리적 데이터 형식을 설계하여 정의한다.

5. 연동 데이터 테이블 매핑 정의

설계된 연동 데이터가 저장/관리될 수 있도록 응용 테이블과 매핑한다.

- 연동 데이터 테이블 매핑 정의
연동 데이터가 저장/관리된 테이블을 설계하고 관련된 응용 데이터를 식별해 연동 테이블과 응용 테이블간의 관계를 정의한다. 또한 연동 데이터와 응용 데이터 테이블간 컬럼 단위로 매핑을 실시하고 세부적인 변환 로직을 설계한다.

작업 수행 지침

■ 연동 항목 설계 작업의 수행 여부

연동 항목 설계 작업은 연동 표준을 정의하고 세부 매핑 로직을 정의해야 하는 작업으로써 이후 연동 구현 작업에 영향을 미치므로 개발 대상 시스템이 연동 소요를 가질 경우에 한해 반드시 실시해야 한다.

■ 코드 자료 매핑 설계시 고려사항

- 시스템간 코드가 불일치하는 경우는 코드 분류 체계가 달라서 별도의 로직이 없으면 매핑이 불가능한 경우가 있을 수 있다. 이러한 경우 해당 로직을 설계하여 산출물에 반영하여야 한다.

■ 연동 데이터 테이블 매핑 정의시 고려사항

- 일반적으로 하나의 연동 항목이 하나의 인터페이스 테이블로 정의되는 경우, 데이터 송신 상황에서 시스템내 응용 데이터가 어떠한 로직을 거쳐 인터페이스 데이터를 생성할 수 있는지 정의되어야 한다.
- 또한 데이터 수신 상황에서 인터페이스 데이터가 수신되면 어떠한 로직을 거쳐 응용 데이터로 변환되고 이들이 각각의 테이블에 삽입되는지를 정의해야 한다.

연동 설계 작업(2D24)

- 연동 데이터와 응용 데이터간의 변환 로직은 응용 소프트웨어로 구현되거나 DBMS의 기능을 이용할 수 있으므로 정의 결과에 따라 관련 산출물을 보완해야 한다.

■ 파일 송수신 방식의 인터페이스 기술

- 일반적으로 파일 송수신 방식의 인터페이스는 파일 형식 정의를 제외하고는 정의해야 할 내용이 많지 않은 대신, 송수신을 위한 모듈 설계가 보다 중요하므로 연동 설계서의 작성 항목을 테일러링하거나 컴포넌트 설계서(2D21a)에 설계 결과를 명시한다.

■ 인터페이스 기술의 사전 정의시의 산출물 작성

- 패키지 소프트웨어나 상용 프레임워크의 도입, 기존 시스템의 커스터마이징, 타 시스템의 연동 모듈 제공 등과 같이 연동 기술 적용에 필요한 별도의 설계 단계 산출물(예 : EAI 네이밍 정의서, EAI 시스템 설계서 등)이 존재할 경우 이를 부록으로 첨부한다.

■ 프로토타입 기법

- 필요시 사용자와 요구 내용의 이해 및 검증, 연동 성능 평가 등의 목적으로 프로토타입을 제작하여 활용할 수 있다.

연동 설계 작업(2D24)

[서식] 연동 설계서(2D24a)

1. 연동 프로그램

1.1. (인터페이스명)

프로그램 구분	프로그램명	설 명

2. 공통 정보 정의

2.1. 연동 메타 정보 테이블 정의

2.1.1. (인터페이스명)

2.1.1.1. (테이블ID)

컬럼명	설명	키 유형	Not Null	데이터 타입 및 길이	제약조건

2.2. 연동 관련 코드 정의

2.2.1. (인터페이스명)

코드ID	코드명	코드값	코드값명	비고

2.3. 오류 메시지 정의

2.3.1. (인터페이스명)

구분	메시지 ID	메시지	오류 설명	대응 방법

연동 설계 작업(2D24)

2.4. 코드 자료 매핑 정의

2.4.1. (인터페이스명)

(개발 대상 시스템 명)				(타 시스템 명)				매핑 규칙
코드 ID	코드 명	코드 값	코드 값명	코드 ID	코드 명	코드 값	코드 값명	

2.5. 연동 테이블 공통 컬럼 정의

2.5.1. (인터페이스명)

컬럼명	설명	키 유형	Not Null	데이터 타입 및 길이	제약조건

2.6. 연동 메시지 형식 정의

2.6.1. (인터페이스명)

2.6.1.1. (메시지명)

순서	그룹	데이터명	데이터 길이	설 명

연동 설계 작업(2D24)

3. 인터페이스 데이터 목록

인터페이스명	연동 항목ID	연동 항목명	주기	처리량	크기	단위	적시성	트랜잭션 타입	트리거 이벤트	보안 등급	상호운용성 달성가능 수준	송신 시스템	수신 시스템	연동 프로그램	1회 전송 건수	전송 제한 시간	메시지간 순차 관계	에러 처리 방법

연동 설계 작업(2D24)

4. 인터페이스 데이터 상세

4.1. (연동 항목ID)

데이터 항목	데이터 타입 및 길이	설 명

연동 설계 작업(2D24)

■ [항목 설명] 연동 설계서(2D24a)

1. 연동 프로그램

연동 과정에서 데이터 송수신 컴포넌트에 데이터를 전달하는 프로그램의 목록을 기술한다.

1.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

- 프로그램 구분
연동 프로그램에 대한 구분(예 : DBMS 프로시저, 함수, 트리거 또는 컴포넌트 및 메소드명 등)을 기재한다.
- 프로그램명
연동 프로그램을 호출할 수 있는 명칭을 기재하되, 필요시 입력 파라미터도 함께 기재한다.
- 설명
연동 프로그램에 대한 상세한 설명을 기술한다.

2. 공통 정보 정의

2.1. 연동 메타 정보 테이블 정의

연동을 관리하기 위한 메타 정보(예 : 송수신현황, 기준정보, 에러, 에러 코드 정의, 기관 코드, 시스템 코드 등)를 저장하는 테이블에 대해 기술한다.

연동에 관한 정보만을 구분하여 식별성을 높이기 위해 산출물을 작성을 권장하지만 만일 데이터베이스 설계서와 통합이 필요하다면 연동 설계서에서는 생략될 수 있다.

2.1.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

2.1.1.1. (테이블ID)

테이블을 식별할 수 있는 식별자를 기재한다.

- 컬럼명
논리 엔티티 속성에 대응되는 컬럼의 명칭을 기재한다.

연동 설계 작업(2D24)

- 설명
컬럼에 대한 간단한 설명 또는 논리 엔티티 속성명을 기재한다.
- 키 유형
컬럼의 키 여부와 유형으로써 주키(PK : Primary Key), 외래키(FK : Foreign Key)를 기재하고 주키일 경우 주키 순서 번호를 함께 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.
- 데이터 타입 및 길이
DBMS에 맞는 컬럼의 데이터 타입 및 길이를 기재한다.
- 제약조건
필요 시 각 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.

2.2. 연동 관련 코드 정의

연동에 관련된 공통 코드(예 : 기관 식별, 시스템 식별, 전송과정, 전송성공여부, 전송목적구분 등)에 대해서 정의하여 기술한다. 확정된 코드는 운용자 지침서에도 반영될 수 있도록 한다.

2.2.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

- 코드ID
코드 그룹을 식별할 수 있는 식별자를 기재한다.
- 코드명
코드 그룹을 대표할 수 있는 명칭을 기재한다.
- 코드값
코드를 구성하는 세부 코드의 실제 값을 기재한다.
- 코드값명
코드값을 대표하는 명칭을 기재한다.
- 비고
코드값에 대한 세부적인 설명 또는 기타 사항을 기술한다.

연동 설계 작업(2D24)

2.3. 오류 메시지 정의

연동시에 발생할 수 있는 오류에 대해 식별하고 이를 코드로 정의한다. 확정된 코드는 운용자 지침서에도 반영될 수 있도록 한다.

2.3.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

- 구분
오류가 발생하는 영역을 구분할 수 있는 명칭을 기재한다.
- 메시지 ID
오류 메시지를 식별할 수 있는 식별자를 기재한다.
- 메시지
오류 발생시 상황을 알려주는 메시지를 기재한다.
- 오류 설명
오류에 대한 상세한 설명을 기술하되, 오류가 발생한 원인, 상세 로그 확인 방법을 포함해서 기술한다.
- 대응 방법
오류에 대한 해소 방법, 재처리 필요 여부 및 방법 등을 기술한다.

2.4. 코드 자료 매핑 정의

개발 대상 시스템과 연동 대상 시스템간 코드 데이터 체계 또는 데이터 자체가 불일치할 경우 이에 대한 매핑을 기술한다.

2.4.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

- (개발 대상 시스템 명)
개발될 시스템을 대표하는 명칭을 기재한다.
- (타 시스템 명)
연동 대상의 타 시스템을 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

연동 설계 작업(2D24)

- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 ID를 기재한다.
- 코드명
코드를 대표하는 명칭을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 명칭을 기재한다.
- 코드값
코드를 구성하는 세부 코드의 실제 값을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 값을 기재한다.
- 코드값명
코드값을 대표하는 명칭을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 명칭을 기재한다.
- 매핑규칙
코드간의 관계 또는 코드를 매핑시키기 위한 별도의 방법을 기술한다.

2.5. 연동 테이블 공통 컬럼 정의

연동 항목별로 테이블을 정의할 경우 공통적으로 들어가는 컬럼(예 : EAI 일련 번호, 전송일시, 전송 구분자 등)을 정의한다.

2.5.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

- 컬럼명
논리 엔티티 속성에 대응되는 컬럼의 대표 명칭을 기재한다.
- 설명
컬럼에 대한 간단한 설명 또는 논리 엔티티 속성명을 기재한다.
- 키 유형
컬럼의 키 여부와 유형으로써 주키(PK : Primary Key), 외래키(FK : Foreign Key)를 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.
- 데이터 타입 및 길이
DBMS에 맞는 컬럼의 데이터 타입 및 길이를 기재한다.

연동 설계 작업(2D24)

- 제약조건

필요 시 각 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부(공통코드, 오류코드) 등을 기술한다.

2.6. 연동 메시지 형식 정의

전문 메시지가 SAM(Sequential Access Method, 순차접근방법) 파일과 같이 정보가 특정 순서로 전송되어야 하는 경우(예 : 헤더에 암호화 키, 메시지 식별자, 버전 정보 등을 포함), 또는 전자 문서 교환 시스템의 경우 DTD(Document Type Definition)와 같이 데이터 구성 요소가 전체적으로 특정 형식을 준수해야 하는 경우, 이에 대한 형식을 정의한다. 필요에 따라 적절히 서식을 변경하여 작성한다.

2.6.1. (인터페이스명)

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서에서 정의된 명칭을 기재한다.

2.6.1.1. (메시지명)

- 순서

메시지를 구성하는 데이터 항목의 순서 또는 일련번호를 기재한다.

- 그룹

구성 데이터 항목이 일정한 그룹을 형성하면 이를 대표하는 명칭을 기재한다. (예 : 메시지 헤더, 풋터)

- 데이터명

구성 데이터 항목을 대표하는 명칭을 기재한다.

- 데이터 길이

구성 데이터 항목의 길이를 기재한다.

- 설명

구성 데이터 항목에 대한 간략한 설명을 기술한다.

3. 인터페이스 데이터 목록*

- 인터페이스명

시스템간의 인터페이스를 대표하는 명칭을 기재하되, 연동 소요 정의서(IER)에서 정의된 명칭을 기재한다.

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

연동 설계 작업(2D24)

- 연동 항목ID

연동되는 정보의 묶음을 식별할 수 있는 식별자를 기재한다.

- 연동 항목명

연동 항목을 대표할 수 있는 명칭을 기재하되, 연동 항목 소요서(IER)에서 식별된 명칭을 기재한다.

- 주기

데이터 인터페이스의 처리주기를 기재한다. 전·평시 주기가 구분되어 있거나 여러 번의 특정 시간에 연동이 된다면 별도란을 이용해 직접 기재한다.

구분	1초 / 1분 / 1시간 / 1일 / 1주일 / 1달 / 1년 / 1년이상 / 이벤트발생시 / 수시로 / 기타(직접기재)
----	---

- 처리량

데이터의 초당 처리량을 기재한다. (예 : 10kb/s, 1mb/s 등)

- 크기

데이터의 크기를 기재한다.

- 단위

데이터의 측정 단위를 기재한다.

구분	BITS / BYTES / KB / MB / GB / TB / 기타(직접기재)
----	---

- 적시성

시스템 또는 시스템 기능을 만족하기 위한 필요한 시간의 정도를 기재한다.

구분	1일 / 1시간 / 1달 / 8시간 / 보통(1-10 초) / 천천히(10 초 ~ 10분) / 근실시간 / 1초미만 / 실시간 / 기타(직접기재)
----	---

- 트랜잭션 타입

연동의 트랜잭션 타입을 공시(Publish), 검색(Search), 동의(Subscribe), 협력(Collaborate)로 구분하여 기재한다.

- 트리거 이벤트

연동을 트리거하는 비즈니스 이벤트에 대한 간단한 설명을 기술한다.

- 보안등급

‘군사보안업무훈령’에 정의된 보안등급을 기재한다.

구분	평문 / 대외비 / III급 / II급 / I급 / SI(특수정보) / 기타(직접기재)
----	--

- 상호운용성 달성가능 수준

시스템이 달성되거나 달성가능한 상호운용성 수준(Levels of Information System Interoperability)을 기재한다.

연동 설계 작업(2D24)

구분	0 : Level 0-하위 수준이 없는 격리된 수준 /
	1 : Level 1-하위 수준이 없는 연결된 수준 /
	1a : Level 1-연결된 수준, 하위수준 a /
	1b : Level 1-연결된 수준, 하위수준 b /
	1c : Level 1-연결된 수준, 하위수준 c /
	1d : Level 1-연결된 수준, 하위수준 d /
	2 : Level 2-하위 수준이 없는 기능적인 수준 /
	2a : Level 2-기능적인 수준, 하위수준 a /
	2b : Level 2-기능적인 수준, 하위수준 b /
	2c : Level 2-기능적인 수준, 하위수준 c /
	3 : Level 3-하위수준이 없는 도메인 수준 /
	3a : Level 3-도메인 수준, 하위수준 a /
	3b : Level 3-도메인 수준, 하위수준 b /
	3c : Level 3-도메인 수준, 하위수준 c /
	4 : Level 4-하위 수준이 없는 엔터프라이즈 수준 /
	4a : Level 4-엔터프라이즈 수준, 하위수준 a /
	4b : Level 4-엔터프라이즈 수준, 하위수준 b /
	4c : Level 4-엔터프라이즈 수준, 하위수준 c /
	기타(직접기재)

- 송신 시스템
연동 정보를 송신하는 시스템을 대표하는 명칭이나 식별할 수 있는 범위 내에서 약어를 기재한다.
- 수신 시스템
연동 정보를 수신하는 시스템을 대표하는 명칭이나 식별할 수 있는 범위 내에서 약어를 기재한다.
- 연동 프로그램
트리거 이벤트가 호출해서 실제 데이터 교환을 유발하는 연동 프로그램명과 파라미터를 기재한다.
- 1회 전송 건수
1회 전송시 같은 형식의 데이터를 묶음으로 전달아서 보내야 하는 건수가 정해져 있을 경우 이를 기재한다.
- 전송 제한 시간
전송이 일정 시간 이내에 완료되지 못하면 예외 처리로 전환되는 한계 시간을 기재한다.
- 메시지 순차 관계
우선 전송되어야 하는 연동 항목ID를 기재한다.

연동 설계 작업(2D24)

- 에러 처리 방법

전송이 실패했을 경우 실패한 데이터 처리에 대한 방법(예 : EAI재전송, 재전송 없음)을 기술한다.

4. 인터페이스 데이터 상세

인터페이스 데이터에 대한 상세 규약을 정의한다. 분석 단계에서 작성된 연동 항목 소요서(IER)의 각 연동 항목에 대한 물리적 데이터 형식을 설계하여 기술한다.

4.1. (연동 항목ID)

인터페이스 데이터 목록의 연동 항목ID와 동일하게 기재한다.

- 데이터 항목

인터페이스 데이터의 세부 항목을 기재한다. 연동 항목 소요서(IER)의 주요 데이터 항목과 동일하다.

- 데이터 타입 및 길이

인터페이스 데이터 항목의 물리적 데이터 타입과 길이를 기재한다. 데이터 타입은 특정 DBMS에 독립적인 형태로 기재해야 하며, 필요에 따라 길이 대신 크기 형태로 기재가 가능하다. (예 : String 20Byte)

- 설명

인터페이스 데이터 항목에 대한 간략한 설명을 기술한다.

연동 설계 작업(2D24)

[서식] 부록 - 연동 항목 테이블 매핑 설계서

1. (연동 항목 ID)

응용 테이블						
식별자	테이블ID	컬럼ID	컬럼명	Not Null	데이터 타입 및 길이	제약조건

연동 테이블						
식별자	테이블ID	컬럼ID	컬럼명	Not Null	데이터 타입 및 길이	제약조건

매핑 방법			
매핑식별자 (순번)	매핑/변환 정보	전송 순순위 식별자	비고

연동 설계 작업(2D24)

■ [항목 설명] 부록 - 연동 항목 테이블 매핑 설계서

1. (연동 항목 ID)

연동되는 정보의 묶음을 식별할 수 있는 식별자를 기재하되, 연동 설계서에 기재된 식별자를 기재한다.

- 응용 테이블

송신할 연동 항목을 생성하는데 필요한 원천 데이터가 저장된 응용 테이블이나 수신한 연동 항목의 값을 저장할 응용 테이블명을 기재한다.

- 연동 테이블

송신할 연동 항목을 저장하거나 수신된 연동 항목이 저장되는 연동 테이블명을 기재한다. 연동 항목과 1:1로 매핑하는 별도의 연동 테이블을 사용하지 않고 응용 테이블에서 직접 연동 항목을 생성하거나 받을 경우, 연동 테이블 서식을 연동 항목으로 테일러링하여 작성한다.

- 식별자

응용 테이블의 각 컬럼과 연동 테이블의 각 컬럼을 식별할 수 있는 식별자를 기재하되, 응용 테이블의 식별자와 연동 테이블의 식별자는 구분해서 기재한다.

- 테이블ID

데이터를 송신 또는 수신하기 위한 대상 테이블의 식별자를 기재한다. 예를 들어 하나의 연동 항목을 송신하기 위해 여러 테이블에서 데이터를 추출해야 한다면 송신 테이블에 각각의 테이블ID를 기재한다. 반대의 경우로써 하나의 연동 항목을 수신해서 여러 테이블에 나뉘어 반영되어야 한다면 수신 테이블에 각각의 테이블 ID를 기재한다.

- 컬럼ID

데이터를 송신 또는 수신하기 위한 대상 컬럼의 식별자를 기재한다. 기재 방법은 테이블ID와 동일하다.

송수신 컬럼을 매핑을 위해 반드시 줄을 맞춰서 작성할 필요는 없다. 매핑 로직은 매핑 방법 정의를 통해 작성한다.

- 컬럼명

데이터를 송신 또는 수신하기 위한 대상 컬럼을 대표하는 명칭을 기재한다.

- Not Null

컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.

- 데이터 타입 및 길이

컬럼의 데이터 타입 및 길이를 기재한다.

연동 설계 작업(2D24)

- 제약조건
필요시 송수신 과정에서 참고해야 할 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.
- 매핑방법
송수신 컬럼에 대한 매핑 방법을 기술한다.
- 매핑 식별자(순번)
매핑 방법을 식별할 수 있는 식별자를 기재한다.
- 매핑/변환 정보
송신 컬럼으로부터 수신 컬럼을 만들기 위한 로직을 기술한다. 기술 방법은 복잡하지 않게 논리식으로 작성하되, 필요시 프로그래밍 언어나 SQL 문장을 이용할 수 있고 함수에 의해 변환되는 경우 함수명을 이용해 기술할 수 있다.
- 전송 선순위 식별자
해당 데이터 항목이 전송되기 전에 우선으로 전송되어야 하는 데이터(예 : FK로 연결된 컬럼)가 있다면 해당 매핑 식별자를 기재한다.
- 비고
기타 부가사항을 기술한다.

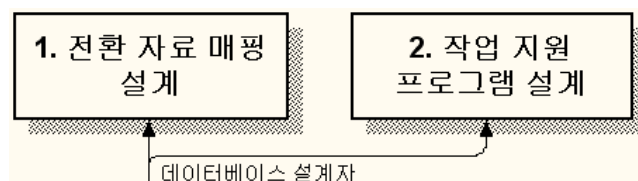
자료 구축 설계 작업(2D25)

개요

구축될 자료를 개발 대상 시스템에 탑재하기 위해 구축 대상 자료와 개발 대상 시스템의 데이터베이스간의 매핑 관계를 설계하고 이 관계를 반영하여 전환 또는 구축 작업을 자동화할 수 있는 프로그램을 설계하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 시스템 아키텍처 정의서 현행 시스템 분석서 자료 구축 계획서 데이터베이스 설계서, (부록)코드 설계서
산출물	<ul style="list-style-type: none"> 자료 구축 설계서
참여자	<ul style="list-style-type: none"> 데이터베이스 설계자 <p>자료를 전환하기 위한 매핑 로직을 설계하고 전환 또는 구축 작업을 효율적으로 실시할 수 있는 프로그램을 설계한다.</p>
관련기법	해당 사항 없음

절차 흐름도



절차 설명

1. 전환 자료 매핑 설계

전환되는 코드와 개발 대상 시스템의 코드가 불일치할 경우 이를 매핑하기 위한 코드 매핑을 정의한다. 코드가 불일치하는 경우는 코드 값이 1:1로 불일치 한 경우와 코드 분류 체계가 달라서 별도의 로직이 없으면 매핑이 불가능한 경우가 있을 수 있다. 이러한 경우 별도의 로직을 설계하여 산출물에 반영하여야 한다. 전환되는 데이터가 개발 대상 시스템의 데이터 형식이나 구조가 정확하게 일치되지 않거나 전환되면서 분산, 통합되어야 할 경우 이에 대한 매핑 관계를 설계한다.

■ 자료 구축 설계 작업(2D25)

2. 작업 지원 프로그램 설계

자료 전환 또는 구축을 자동화할 프로그램(또는 스크립트나 SQL문)이 필요할 경우 이에 대한 설계를 실시한다. 각 자료의 특성과 작업 전략에 맞도록 규칙을 반영하여 작업 지원 프로그램을 설계하고 작업후 결과에 대한 검증할 수 있는 프로그램을 추가로 설계하고 관련 산출물을 첨부한다.

■ 작업 수행 지침

■ 자료 구축 설계 작업의 수행 여부

자료 구축 설계 작업은 자료 매핑이 필요할 경우 이를 정의하고 실제 수행 과정에서 사용되는 스크립트 등이 정밀한 설계가 필요하므로 개발 대상 시스템이 자료 구축 소요를 가질 경우에 한해 반드시 실시해야 한다.

자료 구축 설계 작업(2D25)

[서식] 자료 구축 설계서(2D25a)

1. 전환 코드 매핑

(대상 시스템명)				(개발 대상 시스템명)				매핑 규칙
코드 ID	코드명	코드값	코드값명	코드 ID	코드명	코드값	코드값명	

2.1. (자료 전환 대상명)

[illegible]절차 및 서식 설명 - 239

3.1. (구축 자료 대상명)

[illegible]절차 및 서식 설명 - 240

자료 구축 설계 작업(2D25)

■ [항목 설명] 자료 구축 설계서(2D25a)

1. 전환 코드 매핑

전환 대상 코드와 개발 대상 시스템 코드간 데이터 체계 또는 데이터 자체가 불일치할 경우 이에 대한 매핑을 기술한다.

- (대상 시스템명)
전환 대상 자료를 보유하고 있는 시스템을 대표하는 명칭을 기재한다.
- (개발 대상 시스템명)
개발될 시스템을 대표하는 명칭을 기재한다.
- 코드ID
코드를 유일하게 식별할 수 있는 식별자나 일련번호를 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 ID를 기재한다.
- 코드명
코드를 대표하는 명칭을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 명칭을 기재한다.
- 코드값
코드를 구성하는 세부 코드의 실제 값을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 값을 기재한다.
- 코드값명
코드값을 대표하는 명칭을 기재하되, 개발 대상 시스템의 경우는 코드 설계서에 정의된 명칭을 기재한다.
- 매핑 규칙
코드간의 관계 또는 코드를 매핑시키기 위한 별도의 방법을 기술한다.

2. 전환 자료 매핑

2.1. (자료 전환 대상명)

자료 전환 대상명은 자료를 비즈니스 관점으로 묶어 작성한다. 하나의 대상은 여러 테이블로 구성될 수 있다.

- (대상 시스템명)
전환 대상 자료를 보유하고 있는 시스템을 대표하는 명칭을 기재한다.
- (개발 대상 시스템명)
개발될 시스템을 대표하는 명칭을 기재한다.

자료 구축 설계 작업(2D25)

- 식별자
컬럼을 식별할 수 있는 식별자를 기재하되, 대상 시스템 부분의 식별자와 개발 대상 시스템 부분의 식별자는 구분해서 기재한다.
- 테이블ID
데이터를 전환하기 위한 대상 테이블의 식별자를 기재한다. 예를 들어 하나의 데이터를 전환하기 위해 여러 테이블에서 데이터를 추출해야 한다면 대상 시스템 테이블에 각각의 테이블ID를 기재한다. 반대의 경우로써 하나의 데이터를 전환해서 여러 테이블에 나뉘어 반영되어야 한다면 개발 대상 시스템 테이블에 각각의 테이블 ID를 기재한다.
- 컬럼ID
데이터를 전환하기 위한 컬럼의 식별자를 기재한다. 기재 방법은 테이블ID와 동일하다.
컬럼 매핑을 위해 반드시 줄을 맞춰서 작성할 필요는 없다. 매핑 로직은 전환 방법 정의를 통해 작성한다.
- 컬럼명
데이터를 전환하기 위한 컬럼을 대표하는 명칭을 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.
- 데이터 타입 및 길이
컬럼의 데이터 타입 및 길이를 기재한다.
- 제약조건
필요시 전환 과정에서 참고해야 할 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.
- 전환방법
컬럼에 대한 전환 방법을 기술한다.
- 전환 식별자(순번)
전환 방법을 식별할 수 있는 식별자를 기재한다.
- 매핑/변환 정보
대상 시스템 컬럼으로부터 개발 대상 시스템 컬럼을 만들기 위한 로직을 기술한다. 기술 방법은 복잡하지 않게 논리식으로 작성하되, 필요시 프로그래밍 언어나 SQL 문장을 이용할 수 있고 전환 프로그램에 의해 변환되는 경우 프로그램명을 이용해 기술한다.
- 전환 선순위 식별자
해당 데이터 항목이 전환되기 전에 우선으로 전환되어야 하는 데이터(예 : FK로 연결된 컬럼)가 있다면 해당 전환 식별자를 기재한다.

자료 구축 설계 작업(2D25)

- 비고
기타 부가사항을 기술한다.

3. 초기 구축 자료 매핑

3.1. (구축 자료 대상명)

구축 자료 대상명은 개발 대상 시스템 관점에서 구축되어야 할 자료로 묶어 작성한다. 하나의 대상은 여러 테이블로 구성될 수 있다.

- (원천 자료명)
구축 대상 자료의 원천 자료 명칭을 기재한다.
- 식별자
원천 자료 항목을 식별할 수 있는 식별자를 기재한다.
- 세부 원천 자료명
원천 자료가 여러 자료로 구성되어 있을 경우 이를 구분하기 위한 명칭을 기재한다.
- 원천 자료 항목명
향후 테이블의 컬럼으로 구축될 원천 자료 항목 명칭을 기재한다.
- 제약조건
원천 자료 데이터의 제약조건을 기술한다.
- (개발 대상 시스템명)
개발 대상 시스템을 대표하는 명칭을 기재한다.
- 식별자
컬럼을 식별할 수 있는 식별자를 기재한다.
- 테이블ID
구축 자료가 저장될 대상 테이블의 식별자를 기재한다.
- 컬럼ID
데이터를 구축하기 위한 컬럼의 식별자를 기재한다.
- 컬럼명
데이터를 구축하기 위한 컬럼을 대표하는 명칭을 기재한다.
- Not Null
컬럼의 Null 데이터 허용 여부를 Y, N으로 기재한다.
- 데이터 타입 및 길이
컬럼의 데이터 타입 및 길이를 기재한다.

자료 구축 설계 작업(2D25)

- 제약조건
필요시 구축 과정에서 참고해야 할 컬럼의 제약조건을 기술하되, 허용치에 대한 범위 및 구분값, 기본값(default value), 코드 여부 등을 기술한다.
- 구축방법
컬럼에 대한 자료 구축 방법을 정의한다.
- 구축 식별자(순번)
구축 방법을 식별할 수 있는 식별자를 기재한다.
- 매핑/변환 정보
원천 자료로부터 개발 대상 시스템 컬럼을 만들기 위한 로직을 기술한다. 기술 방법은 복잡하지 작성하되, 필요시 프로그래밍 언어나 SQL 문장을 이용할 수 있고 별도 그림에 의해 변환되는 경우 프로그램명을 이용해 기술한다.
- 비고
해당 데이터 항목을 구축하기 위한 전제조건 및 기타 부가사항을 기술한다.

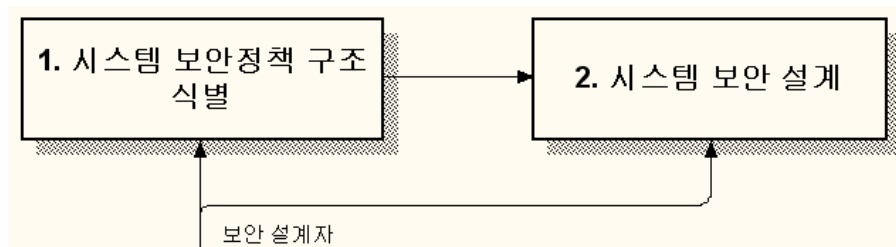
시스템 보안 설계 작업(2D26)

개요

시스템 보안 설계 작업은 시스템에서 관리해야 하는 보안 대상을 보안정책 및 구조 등에 따라 어떻게 적용할 것인가를 정의하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 훈령 및 지침 • 현행 시스템 분석서 • 시스템 아키텍처 정의서
산출물	<ul style="list-style-type: none"> • 시스템 보안 설계서
참여자	<ul style="list-style-type: none"> • 보안설계자 <p>시스템 특성 및 요구사항에 적합하게 시스템 보안정책 구조를 식별하고 그 결과를 설계에 반영한다.</p>
관련기법	해당 사항 없음
상호운용성 관련사항	<ul style="list-style-type: none"> • 국방 아키텍처 프레임워크(MND-AF) v1.2

절차 흐름도



절차 설명

1. 시스템 보안정책 구조 식별

개발 대상 시스템에 적용되는 보안지침 및 정책 구조를 식별한다.

- 시스템 보안정책 구조 식별

개발 대상 시스템 내에 효과적인 보안관리 체계를 적용하기 위하여 보안규정 및 지침과 국방정보기술아키텍처 등을 조사하고 보안영역 유형, 보안요소 및 하부보안요소에 해당되는 항목들을 식별한다.

■ 시스템 보안 설계 작업(2D26)

2. 시스템 보안 설계

식별된 보안정책 구조에 따라 개발 대상 시스템의 보안사항을 설계한다.

- 보안정책 구조에 따른 보안 설계
시스템 보안정책 구조를 토대로 이에 대응되는 적용 방안 또는 구체적인 설계 사항을 도출하고 하드웨어, 소프트웨어 및 통신장비 등과의 관계를 파악한다.
- 기타 보안 설계
시스템 보안정책 구조 외에 별도의 지침이나 가이드 등을 통해 추가적으로 적용되는 보안사항에 대해 식별하고 해당 내용에 대한 적용 방안 또는 설계 사항을 도출한다.

■ 작업 수행시 고려사항

■ 시스템 보안정책 구조 식별 시 고려사항

- 시스템 보안정책 구조 식별 시 국방정보기술아키텍처에서 정의된 보안정책 구조의 적용을 우선적으로 검토한다. 다만, 개발 대상 시스템의 특성에 따라 별도의 보안정책 구조를 수립할 수 있다.

■ 시스템 보안 설계 시 고려사항

- 시스템 보안 설계 시 사용자에게 의해 식별된 요구사항 이외에 추가적으로 필요한 사항에 대해 식별하여 반영해야 한다. 또한, 시스템 보안 설계의 세부 내용은 관련된 산출물을 통해 확인이 가능하여야 한다.

■ 시스템 보안 설계 작업(2D26)

■ [서식] 시스템 보안 설계서(2D26a)

1. 관련 지침

지침명	설 명	분야	제정년월	최근 개정년월	제정기관

2. 시스템 보안 설계

2.1. 보안정책 구조에 따른 보안 설계

보안구분			설 명	적용/설계 내용
보안영역	보안요소	하부 보안요소		

2.2. 기타 보안 설계

시스템 보안 설계 작업(2D26)

■ [항목 설명] 시스템 보안 설계서(2D26a)

1. 관련 지침*

- 지침명
지침을 대표하는 명칭을 기재한다.
- 설명
지침에 대한 설명을 기술한다.
- 분야
지침이 속한 대표적인 분야를 기재한다.

구분	시스템관리 / 보안 / 구매 / 운영관리 / 기타(직접기재)
----	-----------------------------------

- 제정년월
지침이 최초에 제정된 년/월을 기재한다.
- 최근 개정년월
지침이 최근에 개정된 년/월을 기재한다.
- 제정기관
지침을 제정한 기관의 명칭을 기재한다.(예 : 국방부, 방위사업청)

2. 시스템 보안 설계*

2.1. 보안정책 구조에 따른 보안 설계

식별된 보안정책 구조에 대응되는 시스템 보안 설계 내용을 기술한다.

- 보안영역
시스템에서 관리해야 하는 보안대상에 대한 구분을 기재한다.

구분	기술적 보안 / 관리적 보안 / 물리적 보안
----	--------------------------

- 보안요소
기술적, 관리적, 물리적 보안 영역별 주요 보안요소에 대한 명칭을 기재한다.
- 하부 보안요소
보안요소에 따른 하부보안요소 명칭을 기재한다.
- 설명
하부보안요소에 대한 설명을 기술한다.
- 적용/설계 내용
각각의 하부 보안요소와 관계된 시스템의 하드웨어, 소프트웨어 및 통신장비 등의 명칭 또는 시스템에 하부 보안요소에 대응되는 내용이 어떻게 적용/설계되었는지에 대한 설명을 기술한다.

* 국방아키텍처프레임워크(MND-AF) v1.2 유관 항목

시스템 보안 설계 작업(2D26)

2.2. 기타 보안 설계

식별된 보안정책 구조 이외에 기타 지침이나 가이드를 통해 추가된 보안 설계 내용을 기술한다.

시스템 설치 계획 작업(2D31)

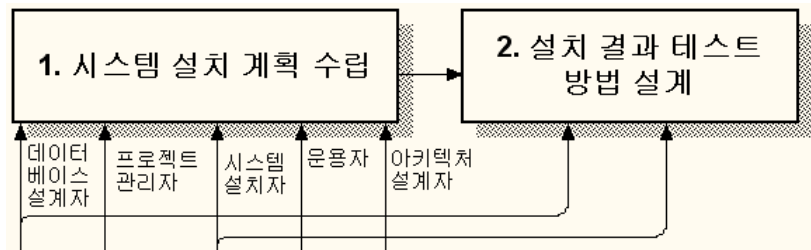
개요

개발 대상 시스템이 구현 완료되면 발주자가 지정한 장소와 플랫폼에 납품자가 이상 없이 시스템을 설치하기 위해 사전에 작업 절차와 일정, 위험 대책 등의 계획을 수립하여 효과적이며 안정된 설치가 될 수 있도록 하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 현행 시스템 분석서 • 시스템 아키텍처 정의서 • 데이터베이스 설계서 • 사용자 인터페이스 설계서 • 컴포넌트 설계서 • 자료 구축 설계서
산출물	<ul style="list-style-type: none"> • 시스템 설치 계획서
참여자	<ul style="list-style-type: none"> • 프로젝트 관리자 상용 품목 납품과 시점을 고려하고 발주자와 협의하여 시스템 설치 계획을 수립하고 납품 검수 관련 행정 절차를 수행한다. • 시스템 설치자 하드웨어, 네트워크, 소프트웨어, 개발된 응용 시스템에 대한 설치 절차 및 위험 대책을 수립한다. • 아키텍처 설계자 설치 대상 시스템이 시스템 아키텍처에 부합하는지에 대한 점검을 수행한다. • 데이터베이스 설계자 데이터 전환 절차 및 위험 대책을 수립한다. • 운전자 설치 일정에 따른 행정절차 및 필요한 제반 도구에 대해 협조한다.
관련기법	해당 사항 없음

시스템 설치 계획 작업(2D31)

절차 흐름도



절차 설명

1. 시스템 설치 계획 수립

하드웨어, 응용 소프트웨어 등 개발 대상 시스템을 운용 환경에 설치하기 위한 설치 계획을 수립한다.

- 시스템 설치 계획 수립

시스템 아키텍처 정의서, 컴포넌트 설계서, 데이터베이스 설계서, 자료 구축 설계서 등을 바탕으로 설치 대상을 정의하고 각 대상별로 설치 일자, 담당자, 절차 및 위험 대책 등을 정의한 면밀한 계획을 수립한다.

2. 설치 결과 테스트 방법 설계

설치 결과에 대한 이상 여부를 판단할 수 있는 테스트 방법을 설계한다.

- 설치 결과 테스트 방법 설계

대상에 대한 설치 작업이 종료하였는지, 시스템이 이상 없이 동작할 수 있는지에 대해 판단할 수 있는 핵심적인 지표를 정의하고 테스트 방법을 설계한다.

예를 들면 데이터 전환 결과는 전환 데이터 건수를 측정하고 네트워크 설치 결과는 관련 도구를 이용하여 네트워크 소통 상태를 측정하며, 응용 소프트웨어 설치 결과는 WAS 에러 메시지 여부, 배포 파일별로 대표적 기능의 호출 및 동작 여부를 테스트할 수 있다.

시스템 설치 계획 작업(2D31)

■ 작업 수행시 고려사항

■ 시스템 설치 계획 작업의 수행 여부

- 시스템 설치 계획 작업은 시스템을 운용 환경에 이상 없이 설치하는데 목적이 있으므로 소프트웨어를 하드웨어에 설치된 상태에서 납품하는 등 해당 사업이 시스템 설치에 대한 과업이 포함되어 있지 않거나 운용 환경에 특별한 설치 작업 요소가 없다면 생략될 수 있다.

■ 시스템 설치 계획 수립시 고려사항

- 개발 대상 시스템이 기존에 운용중인 시스템의 환경 및 운용 방법에 영향을 미친다면 기존 시스템을 정상적으로 운용하면서 신규 시스템을 성공적으로 설치할 수 있도록 계획을 수립한다.
- 예를 들면 비상 전원을 확보하거나 기존 데이터의 백업을 수행하여야 하고 작업 시간을 운용자와 사용자에게 사전에 공지하고 작업 시간 내에는 접속을 차단하거나 예비 시스템을 구동하고 설치 작업 후에는 자료를 일치화하는 등 다양한 상황에 대한 세부적인 작업을 고려해야 한다.
- 만일 기존의 시스템이 신규 시스템으로 전환되는 경우는 별도의 전환 전략이 필요할 수 있다. 전환 방법에는 일반적으로 일시 전환, 병행 전환, 단계적 전환, 파일럿 전환의 4가지 방법이 있다. 각 전환 방법은 각각의 장단점이 존재하므로 적용에 대한 분석이 필요하다.

전환 방법	설 명
일시 전환	특정 시점에 기존 시스템 사용을 완전 중단하고 새로운 시스템을 사용하는 방법이다. 비용이 적게 들고 단시간내 전환이 가능하지만 새로운 시스템에 하자가 있을 경우 업무가 마비되는 위험을 수반할 수 있으므로 이를 위해서는 기존 환경을 백업하고 전환에 대한 성공이 확실할 수 있을 때까지 보관해야 한다. 또한 부가적으로 사용자의 적응상의 문제점이 있을 수 있다.
병행 전환	새로운 시스템의 성능이 확인될 때 까지 기존 시스템과 병행해서 사용하는 방법이다. 위험이 적은 방법이지만 그 만큼 사용자, 운용자 모두가 번거롭고 많은 비용이 요구될 수 있다.
단계적 전환	병행 전환의 일종으로, 새로운 시스템을 하위 시스템으로 나누어 단계적으로 전환해서 성공적일 때 점진적으로 한 부분씩 추가해 나가는 방법이다. 그러나 모든 경우에 기존 시스템을 신규 시스템으로 단계적 전환이 어려운 점이 있다.
파일럿 전환	특정 부서에 시험적으로 가동하여 성공적으로 실행될 경우 전체 조직으로 확산해 나가는 방법이다. 사용자의 저항을 줄일 수 있다.

■ 시스템 설치 계획 작업(2D31)

■ 작업 시점의 조정

- 소프트웨어 개발 공정과 하드웨어 등의 기반 요소 납품 및 설치 시점이 일치하지 않을 수 있으므로 반드시 설계 단계에 시스템 설치 계획 작업을 실시해야 할 필요가 없고 시스템 설치 계획이 수립되어야 하는 적절한 시점에 해당 작업을 진행할 수 있다.
- 최초 시스템 설치 계획 작업 시점에는 현재까지 도출될 수 있는 계획만을 수립하고 향후 공정이 진행됨에 따라 기존 계획을 보완하고 나머지 계획을 추가로 수립할 수 있다.

■ 설치 지원 요소의 설계

- 설치에 필요한 프로그램(또는 스크립트 등) 등 지원 요소가 필요하면 이에 대한 설계를 수행하고 설계 결과를 산출물 부록으로 첨부한다.

■ 지연 대책 수립

- 시스템 설치 요소 중 기 확보되었거나 상시 확보가 보장될 수 없어 시스템 설치 일정에 지장을 줄 수 있는 항목(예를 들어 해외 도입이나 별도 제작중인 장비, 소프트웨어, 하드웨어 등)이 있다면 위험 상황(예 : 제작 지연, 인증 지연 등)에 대한 대책을 수립하여야 한다.

■ 시스템 설치 계획 결과와 이후 작업과의 관계

- 설치 대상 목록, 설치 작업 내용은 시스템 설치 실시 작업(4S11)의 설치 작업으로 유도된다.

시스템 설치 계획 작업(2D31)

[서식] 시스템 설치 계획서(2D31a)

1. 설치 대상 목록

구분1	구분2	이 름
네트워크 통신장비		
하드웨어		
패키지/시스템 소프트웨어		
응용 소프트웨어		
구축 데이터		

2. 설치 작업 내용

2.1. 네트워크 통신장비

2.1.1. (이름)

이름		설치일		담당자	
대상 설명					
설치 내용	수량	설치 장소	네트워크	LAN	
설치 절차					
위험 대책					

2.2. 하드웨어

2.2.1. (이름)

이름		설치일		담당자	
대상 설명					
설치 내용	수량	설치 장소	네트워크	LAN	통신장비
설치 절차					
위험 대책					

시스템 설치 계획 작업(2D31)

2.3. 패키지/시스템 소프트웨어

2.3.1. (이름)

이름	설치일	담당자
대상 설명		
설치 내용	수량	하드웨어
설치 절차		
위험 대책		

2.4. 응용 소프트웨어

2.4.1. (이름)

이름	설치일	담당자
대상 설명		
설치 내용	하드웨어	하드웨어 경로
		배포 파일명
설치 절차		
위험 대책		

2.5. 구축 데이터

2.5.1. (이름)

이름	설치일	담당자
대상 설명		
설치 내용	DBMS	테이블ID
설치 절차		
위험 대책		

시스템 설치 계획 작업(2D31)

■ [항목 설명] 시스템 설치 계획서(2D31a)

1. 설치 대상 목록

설치 대상을 구분하고 이름을 목록으로 작성한다.

- 구분

설치 대상 목록을 묶어 구분(성격, 역할, 설치 작업 등)할 수 있는 구분자를 기재한다.

- 이름

설치 대상을 대표할 수 있는 명칭을 기재한다.

2. 설치 작업 내용

설치 대상에 대한 작업 내용을 기술한다.

2.1. 네트워크 통신장비

2.1.1. (이름)

- 이름

통신장비를 대표하는 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.

- 설치일

설치 예정 날짜를 기재한다.

- 담당자

설치를 담당하게 될 담당자를 기재한다.

- 대상 설명

통신장비에 대한 간략한 설명을 기술한다.

- 수량

설치하게 될 통신장비의 수량을 기재한다.

- 설치 장소

통신장비를 설치하게 될 장소를 기재하되 시스템 노드명으로 기재한다.

- 네트워크

통신장비가 참여하게 될 네트워크 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.

- LAN

통신장비가 참여하게 될 LAN 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.

- 설치 절차

통신장비 설치에 대한 상세한 절차를 기술한다.

시스템 설치 계획 작업(2D31)

- 위험 대책
통신장비 설치로 인한 위험(작업상, 현/타 시스템 운용에 영향 등) 여부를 파악하고 이에 대한 대책을 기술한다.

2.2. 하드웨어

2.2.1. (이름)

- 이름
하드웨어를 대표하는 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.
- 설치일
설치 예정 날짜를 기재한다.
- 담당자
설치를 담당하게 될 담당자를 기재한다.
- 대상 설명
하드웨어에 대한 간략한 설명을 기술한다.
- 수량
설치하게 될 하드웨어의 수량을 기재한다.
- 설치 장소
하드웨어를 설치하게 될 장소를 기재하되 시스템 노드명으로 기재한다.
- 네트워크
하드웨어가 참여하게 될 네트워크 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.
- LAN
하드웨어가 참여하게 될 LAN 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.
- 통신장비
하드웨어가 연결될 통신장비 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.
- 설치 절차
하드웨어 설치에 대한 상세한 절차를 기술한다.
- 위험 대책
하드웨어 설치로 인한 위험(작업상, 현/타 시스템 운용에 영향 등) 여부를 파악하고 이에 대한 대책을 기술한다.

시스템 설치 계획 작업(2D31)

2.3. 패키지/시스템 소프트웨어

2.3.1. (이름)

- 이름
패키지/시스템 소프트웨어를 대표하는 명칭을 기재하되, 시스템 아키텍처 정의서에 기재된 명칭을 기재한다.
- 설치일
설치 예정 날짜를 기재한다.
- 담당자
설치를 담당하게 될 담당자를 기재한다.
- 대상 설명
패키지/시스템 소프트웨어에 대한 간략한 설명을 기술한다.
- 수량
설치할 소프트웨어 수량이나 라이선스 수를 기재한다.
- 하드웨어
패키지/시스템 소프트웨어가 탑재될 하드웨어 명칭을 기재한다.
- 설치 절차
패키지/시스템 소프트웨어 설치에 대한 상세한 절차를 기술한다.
- 위험 대책
패키지/시스템 소프트웨어 설치로 인한 위험(현/타 시스템 운용에 영향 등) 여부를 파악하고 이에 대한 대책을 기술한다.

2.4. 응용 소프트웨어

2.4.1. (이름)

- 이름
응용 소프트웨어를 대표하는 명칭을 기재한다.
- 설치일
설치 예정 날짜를 기재한다.
- 담당자
설치를 담당하게 될 담당자를 기재한다.
- 대상 설명
응용 소프트웨어에 대한 간략한 설명을 기술한다.
- 하드웨어
응용 소프트웨어가 탑재될 하드웨어 명칭을 기재한다.
- 하드웨어 경로
응용 소프트웨어가 탑재될 하드웨어의 디렉토리 경로를 기재한다.

시스템 설치 계획 작업(2D31)

- 배포 파일명
응용 소프트웨어가 하드웨어에 탑재(배포)될 파일명을 기재한다.
- 설치 절차
응용 소프트웨어 설치에 대한 상세한 절차를 기술한다.
- 위험 대책
응용 소프트웨어 설치로 인한 위험(현/타 시스템 운용에 영향 등) 여부를 파악하고 이에 대한 대책을 기술한다.

2.5. 구축 데이터

2.5.1. (이름)

- 이름
구축 데이터를 대표하는 명칭을 기재한다.
- 설치일
설치 예정 날짜를 기재한다.
- 담당자
설치를 담당하게 될 담당자명을 기재한다.
- 대상 설명
구축 데이터에 대한 간략한 설명을 기술한다.
- DBMS
구축 데이터가 탑재될 DBMS를 기재하되, 데이터베이스 설계서에서 기재된 DBMS명을 기재한다.
- 테이블ID
구축 데이터가 삽입될 테이블ID를 기재하되, 데이터베이스 설계서에서 정의된 테이블ID를 기재한다.
- 설치 절차
전환 데이터 설치에 대한 상세한 절차를 기술한다.
- 위험 대책
전환 데이터 설치로 인한 위험(현/타 시스템 운용에 영향 등) 여부를 파악하고 이에 대한 대책을 기술한다.

구현 및 테스트 단계(3T)

개요

구현 및 테스트 단계는 설계 결과를 토대로 개발 대상 시스템을 구현하고 그 결과의 이상 여부와 요구사항 충족 여부에 대해 테스트를 통해 확인하는 것을 목적으로 한다. 구현 및 테스트 단계는 구현, 단위 테스트, 소프트웨어 통합 및 테스트, 시스템 통합 및 테스트, 지침서 작성 활동으로 구성된다.

활동/작업 설명

1. 구현(3T1)

구현 활동은 설계 결과를 토대로 개발 대상 시스템의 데이터베이스, 사용자 인터페이스, 컴포넌트를 구현하여 요구사항을 만족하는 실행 가능한 소프트웨어를 개발한다.

작업ID	작업명	작업내용
3T11	데이터베이스 구축	데이터베이스 설계 결과와 연동 및 자료 구축 설계 결과 등을 토대로 데이터베이스를 구축한다.
3T12	컴포넌트 구현	컴포넌트 설계 결과를 토대로 구현 플랫폼에 최적한 개발 매커니즘(패턴, 스타일)을 적용하여 컴포넌트를 코딩한다.
3T13	사용자 인터페이스 구현	사용자 인터페이스 정의 및 설계 결과를 토대로 사용자 인터페이스를 코딩한다.

구현 및 테스트 단계(3T)

2. 단위 테스트(3T2)

개발 대상 시스템을 구성하는 기본 요소인 컴포넌트, 사용자 인터페이스와 데이터베이스 프로그램 등의 이상 여부를 확인하기 위해 테스트를 수행한다.

작업ID	작업명	작업내용
3T21	단위 테스트 준비	각각의 단위 모듈별로 테스트케이스를 설계하고 수행 절차 및 환경을 준비한다.
3T22	단위 테스트 수행	단위 테스트를 수행하고 발생한 오류를 수정한다. 테스트 결과가 통과 기준을 충족하지 못하면 재 테스트 및 오류수정을 반복한다.

3. 소프트웨어 통합 및 테스트(3T3)

소프트웨어 통합 및 테스트 활동은 소프트웨어 구성요소를 통합하고 이를 검증하기 위해 테스트를 수행한다.

작업ID	작업명	작업내용
3T31	소프트웨어 통합 테스트 준비	소프트웨어 통합 계획을 수립하고 테스트케이스를 설계하며, 수행 절차 및 환경을 준비한다.
3T32	소프트웨어 통합 및 테스트 수행	소프트웨어를 통합한 후, 테스트를 수행하고 발생한 오류를 수정한다. 테스트 결과가 통과 기준을 충족하지 못하면 재 테스트 및 오류수정을 반복한다.

구현 및 테스트 단계(3T)

4. 시스템 통합 및 테스트(3T4)

시스템 통합 및 테스트 활동은 소프트웨어와 하드웨어 등을 통합하고 이를 검증하는 활동으로써 통합된 시스템이 프로젝트의 요구사항을 만족하는지에 대한 테스트를 수행한다.

작업ID	작업명	작업내용
3T41	시스템 통합 테스트 준비	소프트웨어와 하드웨어 등을 통합하기 위한 계획을 수립하고 요구사항 충족여부를 확인할 수 있도록 테스트 케이스를 설계하며, 수행 절차 및 환경을 준비한다.
3T42	시스템 통합 및 테스트 수행	시스템을 통합한 후, 테스트를 수행하고 발생한 오류를 수정한다. 테스트 결과가 통과 기준을 충족하지 못하면 재 테스트 및 오류수정을 반복한다.

5. 지침서 작성(3T5)

개발된 시스템의 기능 및 사용방법을 사용자가 쉽게 이해하고 사용할 수 있도록 사용자 지침서를 작성하고 시스템을 원활하게 운용할 수 있도록 시스템 설치, 관리방법 등을 운용자 지침서에 작성한다.

작업ID	작업명	작업내용
3T51	사용자 지침서 작성	개발된 시스템을 사용자가 쉽게 사용할 수 있도록 시스템 설치 방법 및 공통사항과 기능별 사용법 등에 대해 상세하게 기술한다.
3T52	운용자 지침서 작성	시스템 운용자가 개발된 시스템을 운용할 수 있도록 시스템 구성, 설치 및 관리방법 등에 대한 내용을 기술한다.

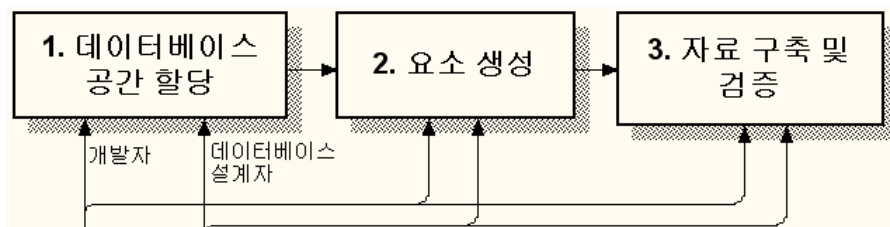
데이터베이스 구축 작업(3T11)

개요

데이터베이스 설계서에 따라 물리적 공간 및 논리적 공간을 할당하고 DDL을 작성하여 이를 DBMS에 등재시키는 작업이다. 또한, DBMS 계층에서 수행될 프로그램(Stored Procedure, Trigger, Sequence 등)을 DBMS에 등재시키며 초기 데이터를 구축하는 작업이다.

구분	작업 구성 항목
입력물	• 데이터베이스 설계서
산출물	• 물리적 데이터베이스
참여자	<ul style="list-style-type: none"> 개발자 데이터베이스 설계자와 함께 물리적 데이터베이스를 구축한다. 데이터베이스 설계자 물리적 데이터베이스 구축 작업을 주도하며 필요시 관리자 역할을 수행하여 데이터베이스 형상을 관리한다.
관련기법	• 지속적 통합

절차 흐름도



절차 설명

1. 데이터베이스 공간 할당

자료 저장량을 산정하고, DB의 구성 요소들이 실행 환경에서 필요한 물리적 공간을 할당하며 논리 공간을 분할한다.

- 자료 저장량 산정
실 데이터의 크기, 발생 빈도, 저장 연한 등을 기초로 자료 저장량을 산정한다.
- 물리 공간 할당
산정한 자료량을 합산하여 전체 공간을 할당한다.

데이터베이스 구축 작업(3T11)

- 논리 공간 분할
물리 공간 할당에서 언급한 정보군 특성 및 컴포넌트 패키지 구조를 참조하여 설정한다. 논리 공간 분할은 DB 보안적 측면에서도 유용하다.

2. 요소 생성

DDL, 프로시저, 트리거, 시퀀스 등 데이터베이스 요소를 생성하여 DBMS에 등재한다.

- 계정 및 권한 생성
각 사용자 계정을 생성하고 이에 대한 권한을 DCL(Data Control Language) 또는 모델링 도구를 이용하여 설정한다.
- 객체 생성
각 객체(테이블, 인덱스, 시퀀스 등)를 DDL(Data Definition Language) 또는 모델링 도구를 이용하여 생성한다.
- 데이터베이스 프로그램 생성
각 데이터베이스 프로그램(뷰, 패키지, 프로시저, 함수, 트리거, 스케줄 등)을 생성한다.

3. 자료 구축 및 검증

자료 구축 계획서와 자료 구축 설계서에 따라 전환 또는 신규 구축한 자료를 DB에 구축 또는 이관하고 그 결과에 대해 검증한다.

- 자료 구축
계획된 자료 전환 및 신규 구축을 수행하고 검증을 실시한다. 검증 결과로써 전환 및 구축에 실패한 자료 발생 등 예외 사항이 발생시에는 별도의 대책을 수립한다.

■ 작업 수행시 고려사항

■ 산출물 제출 방법

- 물리적 데이터베이스 산출물이란 DBMS내에 각 스키마를 생성하고 초기 데이터를 구축하여 이를 하드웨어에 탑재한 상태를 말한다.
- 또한 이와 별도로 계정과 권한, 객체, 데이터베이스 프로그램 등을 생성할 수 있는 DCL/DDDL 스크립트 및 초기 데이터를 생성할 수 있는 DML(Data Manipulation Language) 스크립트를 파일 형태로 제출한다.

데이터베이스 구축 작업(3T11)

■ 공간 할당시 고려사항

- 공간 할당 정책은 DB 튜닝적 입장이 강하며 DBMS의 특성에 의존적이다.
- 대체로 일방적으로 증가하는 정보군(거래, 계약 정보 등), 빈번히 수정되는 정보군(개인 이력, 상품 특성 등), 초기 설정 후 변경과 증가가 좀처럼 발생하지 않는 정보군(우편번호, 직급체계 등), DB 인덱스 설정용 공간, Log 처리용 등 정보군의 특성을 고찰하여 성능 향상을 기할 수 있다.

데이터베이스 구축 작업(3T11)

[서식] 물리적 데이터베이스(3T11a)

해당 사항 없음.

[항목 설명] 물리적 데이터베이스(3T11a)

해당 사항 없음.

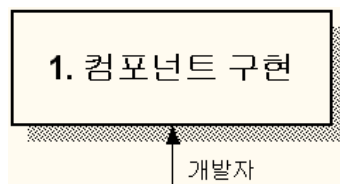
컴포넌트 구현 작업(3T12)

개요

컴포넌트 설계 결과를 바탕으로 컴포넌트 내부 구성 요소들을 플랫폼에 맞게 구현한다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 컴포넌트 아키텍처 명세서 컴포넌트 설계서 트랜잭션 설계서 연동 설계서, (부록)연동 항목 테이블 매핑 설계서 물리적 데이터베이스
산출물	<ul style="list-style-type: none"> 컴포넌트 코드
참여자	<ul style="list-style-type: none"> 개발자 <p>컴포넌트 설계 내용을 바탕으로 컴포넌트를 플랫폼과 코딩 표준에 맞게 구현한다.</p>
관련기법	<ul style="list-style-type: none"> 리팩토링 지속적 통합 작 프로그래밍 코드 리뷰 테스트 주도 개발

절차 흐름도



절차 설명

1. 컴포넌트 구현

컴포넌트 설계 결과를 기반으로 컴포넌트를 구현한다.

- 클래스 구현

컴포넌트 설계 결과를 기반으로 컴포넌트 내부 클래스와 기타 클래스를 구현한다.

컴포넌트 구현 작업(3T12)

- 컴포넌트 생성
구현된 클래스를 구현 플랫폼의 패키징 방식에 따라 컴포넌트별로 묶어 컴포넌트를 생성한다.
- 컴포넌트 등록 및 클래스 배치
컴포넌트를 구현 플랫폼의 등록 방식에 따라 등록하고 기타 클래스를 패키징해 물리적 위치에 배치한다.

■ 작업 수행시 고려사항

■ .Net 기반의 컴포넌트 구현

- .Net 기반 언어(C#.Net, ASP.Net, Visual Basic.Net 등)와 클래스 라이브러리를 선정하여 컴포넌트를 생성한다.
- 프로젝트 안에는 네임스페이스를 반드시 참조하여야 한다. 특히 컴포넌트를 COM+에 등록하기 위해서 서버와 클라이언트 모두에 컴포넌트 생성시 참조되는 중요한 네임스페이스인 System.EnterpriseServices를 추가한다. 그리고 COM+서비스를 받고자 하는 클래스를 ServicedComponent로부터 상속 받는다. 컴포넌트에 클래스와 메소드를 구현하여 환경에 맞게 코딩한다.
- StrongNameKey는 라이브러리 어플리케이션의 경우에는 필요하지 않으나 서버 어플리케이션의 경우에는 전역 어셈블리 캐시(Global Assembly Cache)에 등록하기 위해서 반드시 필요하므로 이를 등록해야 한다.
- .Net 프레임워크가 공통언어 런타임을 인식할 수 있도록 공통언어 런타임 어셈블리에 컴포넌트의 타입을 정의한다.
- 구현된 컴포넌트를 COM+에 등록하고 선택사항으로 등록된 COM+ 컴포넌트를 웹 서비스로 변환할 수 있다.
- 어셈블리의 StrongNameKey로 어셈블리를 빌드하고 디지털 서명을 한다. 생성된 .snk 파일정보는 컴포넌트 생성시 자동 생성된 AssemblyInfo.cs 파일에 .snk 파일명과 그 파일 위치를 명시한다. VB.NET에서는 자동으로 .snk파일을 생성하나 C#의 경우에는 개발자가 직접 작업을 해주어야 한다.
- 컴포넌트의 배포 파일명인 .DLL파일을 COM+에는 패키지 형태로 등록하게 되는데 서비스 컴포넌트를 계승한 클래스 단위로 나타나게 된다. 즉 컴포넌트를 COM+에 등록 시 .DLL로 묶어서 지정한 패키지명을 가지고 등록한다. 컴포넌트 등록 후에는 컴포넌트를 관리하는 '구성요소 서비스' 관리 툴에서 등록된 컴포넌트와 이를 구성하는 클래스들을 볼 수 있다.
- 컴포넌트를 COM+에 등록 후 이를 전역 어셈블리 영역에 등록하고자 한다면 닷넷의 가상 머신(virtual machine)이 쉽게 찾을 수 있도록 전역 어셈블리 캐시에 컴포넌트를 등록한다.

컴포넌트 구현 작업(3T12)

■ J2EE 기반의 컴포넌트 구현

- 컴포넌트 구현 설계서를 바탕으로 엔티티 빈, 세션 빈, 메시지 드리븐 빈을 구현하고 빈과 관련된 홈 인터페이스, 리모트 인터페이스, 로컬 인터페이스를 구현한다. 또한 엔티티 빈일 경우 엔티티 빈의 주요 키 클래스를 구현한다. 주요 키 클래스는 모든 엔티티 빈을 다르게 식별할 수 있도록 해주는 역할을 한다.
- 구현 컴포넌트 내부에서 활용되는 각종 유틸리티 등 관련 클래스를 구현한다. 또한 빈 및 클래스의 각종 메소드에 알고리즘을 구현하고 코딩된 것을 컴파일 하여 문법 오류를 수정한다.
- 컴포넌트 구현 설계서의 배포 정의에 정의된 배포 정보에 따라 컴포넌트 배치 기술서(Deployment Descriptor)를 만들고 관련 빈들과 함께 패키징한다. 배치 기술서는 XML로 작성되며 EJB Spec에 정의된 문법을 반드시 준수해야 한다. 선택사항으로 컴포넌트를 웹 서비스로 변환할 수 있다.
- 홈 인터페이스는 EJB 서버에 있는 컨테이너에게 필요한 정보를 제공하는데 EJB 객체를 생성, 삭제, 검색(Finder Method)하는 메소드를 정의하고 있다. 만약 같은 서버 안에서의 EJB 객체의 생성, 삭제, 검색 메소드를 호출할 때는 로컬 홈 인터페이스를 사용한다.
- 리모트 인터페이스는 EJB를 구현한 빈 클래스에서의 비즈니스 로직 메소드의 선언들을 가지는데 이를 통해 클라이언트는 원격에 있는 빈에 정의되어진 비즈니스 메소드를 호출할 수 있다. 만약 같은 서버 안에서의 비즈니스 로직 메소드를 호출할 때는 로컬 인터페이스를 사용한다.
- 배치 기술서(Deployment Descriptor) 파일에서 빈 클래스, 홈/리모트 인터페이스(로컬 홈/로컬 인터페이스), 기본 키 클래스 이름이 기술되고 데이터베이스 연결에 관련된 설정과 기타 환경설정과 관련된 내용이 포함된다. 또한, 각 트랜잭션에 포함되어야 할 메소드들에 대해 트랜잭션 설계서를 참조하여 트랜잭션 속성을 설정한다.
- 엔터프라이즈 빈의 구조와 외부 엔터프라이즈 빈의 종속관계를 ejb-jar 파일로 표현한다. 구조적인 정보는 외부 엔터프라이즈 빈의 기능과 밀접한 상관관계를 지니고 있기 때문에 반드시 작성되어야 하며, 일반적으로 변경 불가능하다.
- ejb-jar 파일 내에 존재하는 엔터프라이즈 빈이 어떻게 보다 큰 어플리케이션 배치 단위 내에서 구성되어야 하는지를 기술한다. 어셈블리 정보는 선택 사항으로 엔터프라이즈 빈의 기능적인 손상 없이 어플리케이션의 행위를 변경 가능하게 한다.
- 빈 단위 또는 컴포넌트 단위마다 하나의 JAR 파일을 생성해 패키징한다. 빈마다 생성하는 경우에는 컴포넌트를 이루는 빈들의 JAR 파일의 집합이 서버에 전개되어 하나의 컴포넌트로서의 기능을 한다.
- EJB 컴포넌트 개발자에 의해서 생성된 빈 클래스와 홈/리모트 인터페이스(로

컴포넌트 구현 작업(3T12)

컬 홈/로컬 인터페이스), 기본 키 클래스 등을 JAR파일 하나로 묶어준다.
컴포넌트 단위마다 생성하는 경우에는 여러 개의 빈과 클래스들을 하나의 JAR파일로 묶어서 한 번에 서버에 올린다.

■ 코드 주석(annotate) 첨부

- 코드에는 개발자 자신과 타 개발자가 코드를 이해할 수 있도록 주석이 첨부되어야 한다. 적절한 주석은 정도는 프로그램 성격에 따라 각각 다르므로 다음을 고려하여 표준 개발 지침이 수립되어야 하고 각 코드는 지침의 표준을 준수해야 한다.
 - 해당 코드 파일이 무슨 기능을 위한 코드인지를 설명해야 한다.
 - 해당 오퍼레이션이 무슨 기능을 위한 코드인지를 설명해야 한다.
- 너무 많은 주석은 오히려 가독성을 떨어뜨린다. 궁극적으로는 주석이 필요 없는 코드를 만들어야 한다. 즉, 주석 없이 코드만으로도 이해될 수 있도록 변수명이 무슨 역할을 하는지 식별될 수 있어야 하고 코드 자체가 간결해야 한다.

■ 표준 지침의 갱신

- 컴포넌트 구현 시 표준 지침에서 정의한 명명규칙, 주석지침 등의 개발 표준을 준수하여야 한다.
- 구현 작업을 수행하다보면 미처 표준이 정의되지 않았거나 갱신되어야 하는 경우가 발생한다. 개발자는 임의대로 구현하는 것을 지양하고 표준이 수립되어야 하는 필요를 인식할 경우 표준 작성자와 협의하여 표준을 갱신해야 한다.

■ 테스트 주도 개발 적용시 수행 방법

- 테스트 주도 개발(TDD : Test Driven Development) 방법을 적용할 경우에는 단위 테스트(3T2) 작업을 구현(3T1) 작업과 동시에 실시할 수 있다.

■ 산출물 제출 방법

- 컴포넌트 코드란 서버 컴포넌트의 경우, 서버 하드웨어의 WAS 환경에 바이너리 파일을 탑재하는 것을 말하며, 클라이언트 화면의 경우 인스톨 본과 자동 인스톨 환경(예 : 서버 접속시 자동 다운로드 및 인스톨)을 말한다.
- 또한 이와 별도로 서버와 클라이언트에 설치되는 코드 원본(컴파일 이전 코드)을 파일 형태로 제출한다.

컴포넌트 구현 작업(3T12)

[서식] 컴포넌트 코드(3T12a)

해당 사항 없음.

[항목 설명] 컴포넌트 코드(3T12a)

해당 사항 없음.

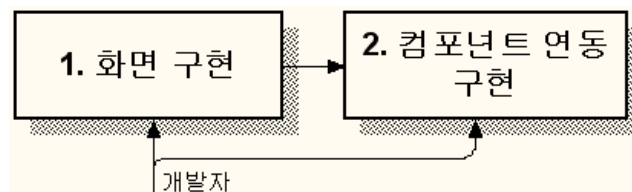
■ 사용자 인터페이스 구현 작업(3T13)

■ 개 요

사용자 인터페이스 정의 및 설계 결과를 바탕으로 화면을 구현하고 구현된 화면을 컴포넌트와 연동시키는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 시스템 아키텍처 정의서 • 사용자 인터페이스 정의서 • 사용자 인터페이스 명세서 • 사용자 인터페이스 설계서 • 시스템 보안 설계서 • 컴포넌트 코드 • 물리적 데이터베이스
산출물	<ul style="list-style-type: none"> • 사용자 인터페이스 코드
참여자	<ul style="list-style-type: none"> • 개발자 <p>사용자 인터페이스 정의 및 설계 결과를 바탕으로 화면을 표준에 맞게 구현한다.</p>
관련기법	<ul style="list-style-type: none"> • 리팩토링 • 지속적 통합 • 짝 프로그래밍 • 코드 리뷰 • 테스트 주도 개발

■ 절차 흐름도



■ 절차 설명

1. 화면 구현

사용자 인터페이스 정의 및 설계 결과를 바탕으로 시스템에서 구현할 화면을 개발 언어에 맞게 구현한다.

■ 사용자 인터페이스 구현 작업(3T13)

- 웹 페이지 구현
HTML 페이지 및 JSP, ASP, PHP 등의 서버 페이지 언어를 이용해 사용자 화면을 구현하고 프로그램 로직이 반영되어 있는 스크립트를 작성한다.
- 웹 컴포넌트 설정
화면과 보고서 등의 UI를 구현하는 컴포넌트가 존재하면 관련 파라미터를 설정하고 리포팅 도구 등을 통해 표현되는 화면 내역 및 보고서 양식을 생성한다.

2. 컴포넌트 연동 구현

서버 컴포넌트와 연결시키는 부분을 코딩한다.

- 웹 클래스 구현
웹 페이지의 처리 흐름 및 사용자 화면과 서버와의 연결을 제어하는 웹 클래스를 구현한다.
- 웹 구성요소 패키징 및 배치
구현된 웹 구성요소를 패키징하고 물리적 위치에 배치한다.

■ 작업 수행시 고려사항

■ 웹 페이지 작성시 고려사항

- 웹 클라이언트의 경우 재사용성과 보안을 고려하여 정적인 HTML 페이지 대신 JSP, ASP, PHP 등의 서버 페이지 언어를 이용하여 동적으로 화면을 구현하고 프로그램 로직이 반영되어 있는 스크립트는 페이지에서 분리한다.
- 데이터 정합성 확보와 프로그램의 간결성을 위해 화면 층에서 입력 필드의 유효값 및 필수 여부 체크, 버튼 제어, 간단한 계산 등의 로직을 구한다.

■ 코드 주석(annotate) 첨부

- 코드에는 개발자 자신과 타 개발자가 코드를 이해할 수 있도록 주석이 첨부되어야 한다. 적절한 주석은 정도는 프로그램 성격에 따라 각각 다르므로 다음을 고려하여 표준 개발 지침이 수립되어야 하고 각 코드는 지침의 표준을 준수해야 한다.
 - 해당 코드 파일이 무슨 기능을 위한 코드인지를 설명해야 한다.
 - 해당 오퍼레이션이 무슨 기능을 위한 코드인지를 설명해야 한다.
- 너무 많은 주석은 오히려 가독성을 떨어뜨린다. 궁극적으로는 주석이 필요 없는 코드를 만들어야 한다. 즉, 주석 없이 코드만으로도 이해될 수 있도록 변수명이 무슨 역할을 하는지 식별될 수 있어야 하고 코드 자체가 간결해야 한다.

■ 사용자 인터페이스 구현 작업(3T13)

■ 표준 지침의 갱신

- 사용자 인터페이스 구현 시 표준 지침 작업에서 정의한 개발 표준을 준수하여야 한다.
- 구현 작업을 수행하다보면 미처 표준이 정의되지 않았거나 갱신되어야 하는 경우가 발생한다. 개발자는 임의대로 구현하는 것을 지양하고 표준이 수립되어야 하는 필요를 인식할 경우 표준 작성자와 협의하여 표준을 갱신해야 한다.

■ 테스트 주도 개발 적용시 수행 방법

- 테스트 주도 개발(TDD : Test Driven Development) 방법을 적용할 경우에는 단위 테스트(3T2) 작업을 구현(3T1) 작업과 동시에 실시할 수 있다.

■ 산출물 제출 방법

- 사용자 인터페이스 코드란 서버 페이지의 경우, 서버 하드웨어의 WAS 또는 웹 서버 프로그램 환경에 코드 또는 바이너리 파일을 탑재하는 것을 말하며, 클라이언트 화면의 경우 인스톨 본과 자동 인스톨 환경(예 : 서버 접속시 자동 다운로드 및 인스톨)을 말한다.
- 또한 이와 별도로 서버와 클라이언트에 설치되는 코드 원본(컴파일 이전 코드)을 파일 형태로 제출한다.

사용자 인터페이스 구현 작업(3T13)

☐ [서식] 사용자 인터페이스 코드(3T13a)

해당 사항 없음

☐ [항목 설명] 사용자 인터페이스 코드(3T13a)

해당 사항 없음

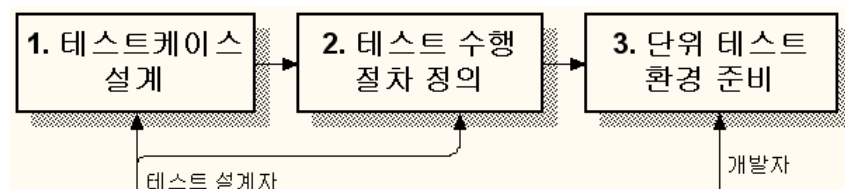
단위 테스트 준비 작업(3T21)

개요

개발 대상 시스템을 구성하는 기본 요소인 컴포넌트와 사용자 인터페이스, 데이터베이스 프로그램 등 단위 소프트웨어의 숨어 있는 결함을 찾기 위해 테스트를 설계하고 제반 사항을 준비하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 요구사항 정의서 사용자 인터페이스 명세서 컴포넌트 설계서 사용자 인터페이스 설계서 데이터베이스 설계서 연동 설계서 테스트 계획서 물리적 데이터베이스 컴포넌트 코드 사용자 인터페이스 코드
산출물	<ul style="list-style-type: none"> 단위 테스트 기술서
참여자	<ul style="list-style-type: none"> 테스트 설계자 테스트 계획서를 바탕으로 테스트케이스와 수행 절차를 설계한다. 개발자 테스트 수행을 위한 데이터를 개발한다.
관련기법	<ul style="list-style-type: none"> 경험 기반 테스트 구조 기반 테스트 명세 기반 테스트

절차 흐름도



단위 테스트 준비 작업(3T21)

절차 설명

1. 테스트케이스 설계

테스트 계획서에서 정의된 테스트 유형에 따라 각 테스트 단위별 테스트케이스와 예상 결과(Test Oracle)를 작성한다.

- 테스트케이스 설계
테스트케이스 설계 기준을 고려해 테스트케이스를 설계한다.

2. 테스트 수행 절차 정의

각 테스트케이스 별로 입력에 따른 예상 출력을 얻기 위한 절차를 정의한다.

- 테스트 수행 절차 정의
테스트를 수행하는 순서 및 사전/제약조건을 정의한다.

3. 단위 테스트 환경 준비

단위 테스트를 수행하기 위한 환경을 준비한다.

- 테스트 프로그램 작성
컴포넌트나 사용자 인터페이스, DB 프로그램의 외부 인터페이스 테스트를 위한 테스트 드라이버(Test Driver)나 테스트 스텝(Test Stub) 등이 필요할 경우 이를 제작한다. 테스트 드라이버란 테스트를 지원하기 위한 스크립트 코드나 데이터로써 대상 컴포넌트의 동작을 호출한다. 테스트 스텝은 테스트 대상 컴포넌트가 호출하여 인터페이스할 더미(dummy) 컴포넌트에 해당한다.
- 데이터베이스 연결
데이터베이스와 연동되는 컴포넌트의 경우 데이터베이스 연결을 준비한다.
- 테스트 데이터 준비
입력값, 조회값 등 테스트케이스 정의에 따라 테스트에 활용될 데이터를 충분히 준비한다.

단위 테스트 준비 작업(3T21)

■ 작업 수행시 고려사항

■ 단위 테스트 준비 작업의 수행 여부

- 단위 테스트 준비 작업은 구현된 단위 소프트웨어의 테스트를 설계하고 환경을 준비하는 것으로써 반드시 실시해야 하나 일반적으로 컴포넌트 구현, 사용자 인터페이스 구현, 데이터베이스 구축 활동과 병행해서 실시할 수 있으므로 소규모 사업의 경우 별도로 분리해서 실시하지 않을 수 있다.

■ 테스트케이스 설계시 고려사항

- 테스트케이스를 생성하는 방법에는 다음과 같은 접근 방법이 있다.
 - 테스트케이스 생성 도구 이용
 - 테스트 데이터 생성 도구 이용 또는 프로그램 개발
 - 유스케이스 시나리오로부터 생성
 - 각 설계서의 명세로부터 생성
- 테스트케이스는 다음 사항을 고려하여 설계한다.
 - 예외상황, 비정상적인 조건에서의 기능 테스트를 포함해야 한다. 예를 들면 대상 데이터가 하나도 없는 경우, 비정상적인 데이터가 입력되는 경우, underflow, overflow 상황 등이다.
 - 수행 경로 시험을 포함해야 한다. 동치분해에 의한 경계값 및 오류(Invalid) 데이터 입력 테스트를 포함해야 한다. 예를 들어 현금 지급기의 지급액 범위가 30만원 이상 1,000만원 이하일 경우, 입력값을 29, 30, 31, 999, 1,000, 1,001, 35.5, -10, Null, 문자, 특수기호 등으로 입력하는 각각의 테스트케이스를 설계한다.
 - 계획된 테스트 커버리지(coverage)를 만족해야 한다.
- 컴포넌트 테스트는 다음을 확인하는 테스트케이스 유형을 포함하는 것을 고려한다.
 - 컴포넌트 내 클래스별 단위 메소드의 동작을 확인
 - 단위 메소드별로 입력에 따른 예상 출력 형태 확인
 - 클래스 내 속성에 대해 적절한 갱신을 확인
 - 모든 입력 필드 타입의 유효성을 확인
 - 가능한 컴포넌트 내 클래스의 모든 코드 라인의 수행(테스트 커버리지)
 - 가능한 모든 분기의 수행(테스트 커버리지)
 - 메소드 호출 관계에 따른 기능성을 확인

단위 테스트 준비 작업(3T21)

- 객체의 상태 변화를 일으키는 임의의 메소드 호출 순서에 따른 기능성을 확인
 - 상속에 따른 클래스의 기능성을 확인
 - 연관 관계에 따른 클래스의 기능성을 확인
 - 컴포넌트의 각 인터페이스에 대한 기능성을 확인
- 사용자 인터페이스 테스트는 다음을 확인하는 테스트케이스 유형을 포함하는 것을 고려한다.
 - 각 버튼의 동작과 링크 상태를 확인
 - 사용자 조작에 대한 정확성 체크 로직을 확인
 - 입력 필드의 유효성 체크 및 필수 입력 체크 로직 확인
 - 각 레이블의 일치성 확인
 - 기타 사용 편리성 확인
 - 데이터베이스 프로그램 테스트는 다음을 확인하는 테스트케이스 유형을 포함하는 것을 고려한다.
 - 내부 동작 로직을 확인
 - 예외 상황 처리 로직을 확인

■ 테스트 수행 절차 정의시 고려사항

- 테스트케이스간 의존성이 있을 경우 이러한 사항을 사전조건으로 정의한다.
- 일반적 수행 절차 외에도 다음의 사항을 고려해야 한다.
 - 테스트를 몇 회 실시할 것인지
 - 어떤 조건으로 실시할 것인지
 - 어떻게 초기화할 것인지
 - 어떤 기술을 이용할 것인지
 - 어떻게 측정할 것인지
 - 결과를 기록하기 위한 특별한 방법이 있는지
 - 어떤 조건이 되면 종료할 것인지 등이다.
- 단위 테스트의 경우 수행 절차가 복잡하지 않다면 생략할 수 있다.

■ 테스트 데이터 작성시 고려사항

- 테스트 데이터는 무작위로 생성된 garbage 데이터와 실제와 유사 또는 동등한 데이터로 나눌 수 있다.

단위 테스트 준비 작업(3T21)

- 일반적으로 실제와 유사한 데이터를 이용해서 테스트를 수행하지만 garbage 데이터를 단순 데이터 저장 및 전송 동작의 확인으로 충분한 경우, 볼륨 및 스트레스 테스트 수행, 오류 상황 연출 등의 목적을 위해 활용될 수 있다.

■ 테스트 주도 개발 적용시 수행 방법

- 구현 활동에서 테스트 주도 개발을 적용하였고 단위 테스트 활동과 구현 활동의 테스트케이스가 일치한다면 이로써 대체할 수 있다.

■ 테스트 종류의 구분과 적용

- 테스트는 크게 어플리케이션을 기동 여부에 따라 정적 테스트와 동적 테스트로 구분된다. 정적 테스트는 Inspection, Review, Walkthrough 등이 있는데, 이는 컴포넌트 구현(3T12) 작업과 동시에 실시되고 각 테스트 활동에서는 해당되지 않는다.
- 동적 테스트에는 내부 로직을 검사하는 White Box 테스트와 외부 기능을 검사하는 Black Box 테스트로 나뉜다. 예를 들어 단위 메소드 테스트, 메소드 상호관계 테스트, 클래스 상호관계 테스트 등을 White Box 테스트 기법으로, 컴포넌트 인터페이스 테스트를 Black Box 테스트 기법으로 수행할 수 있다.

■ 산출물 작성시 고려사항

- 테스트 준비와 수행 작업은 동일한 대상을 이용해 연속적으로 수행되나 사업관리 관점에서 테스트케이스의 완전성을 확인/검증하기 위한 활동(테스트 준비 검토(TRR : Test Readiness Review) 등)을 고려해 두개의 작업으로 분리되었다.
- 본 준비 작업에서는 테스트케이스 및 수행 절차를 설계하고 테스트를 위한 환경을 준비하는데 초점을 두어 해당 내용만을 산출물에 작성하고 수행 작업에서는 수행 결과를 정리해 동일 산출물을 보완한다.

■ 단위 테스트 준비 작업(3T21)

■ [서식] 단위 테스트 기술서(3T21a)

1. (단위) 테스트

1.1. 테스트 설계

1.1.1. (단위명)

1.1.1.1. 테스트케이스

테스트 유형	테스트 항목	테스트 케이스ID	테스트 케이스명	입력값	사전조건	예상 결과

1.1.1.2. 테스트 수행 절차

테스트케이스ID	테스트 수행 절차

단위 테스트 준비 작업(3T21)

■ [항목 설명] 단위 테스트 기술서(3T21a)

1. (단위) 테스트

컴포넌트 테스트, 사용자 인터페이스 테스트, 데이터베이스 프로그램 테스트 등 테스트 유형을 기재한다.

1.1. 테스트 설계

1.1.1. (단위명)

테스트 단위의 명칭을 기재한다. 컴포넌트 테스트의 경우 컴포넌트명을, 사용자 인터페이스 테스트의 경우 화면 ID를, 데이터베이스 프로그램의 경우 프로시저명, 함수명 등을 기재한다.

1.1.1.1. 테스트케이스

- 테스트 유형

테스트 계획서의 테스트 유형을 참고하여 단위 테스트 대상에 해당하는 테스트 유형을 기재한다. 컴포넌트 테스트에서는 단위 메소드 테스트, 메소드 상호관계 테스트, 클래스 상호관계 테스트, 컴포넌트 인터페이스 테스트 등이 될 수 있다.

사용자 인터페이스 테스트의 경우 유효성 테스트, 오퍼레이션 테스트, 사용성 테스트 등이 될 수 있다.

데이터베이스 프로그램 테스트의 경우 동작 테스트 등이 될 수 있다.

- 테스트 항목

테스트를 실제로 수행하는 단위를 기재한다.

- 테스트케이스ID

각 테스트케이스마다 유일하게 구분할 수 있는 식별자를 정의하여 기재한다.

- 테스트케이스명

각 테스트 항목에서 테스트 조건에 따라 분류되는 테스트케이스명을 정의하여 기재한다.

- 입력값

해당 테스트케이스를 위한 구체적인 입력 값을 기재한다.

- 사전조건

각 테스트케이스를 시작하기 전에 미리 갖추어야 할 조건이 있을 경우 이를 기술한다.

- 예상 결과

해당 테스트케이스를 수행할 때 입력 값에 의해 기대되는 결과를 기술한다.

단위 테스트 준비 작업(3T21)

1.1.1.2. 테스트 수행 절차

테스트케이스를 포함하여 테스트를 어떻게 수행할 것인지를 기술하며 테스트의 수행 절차가 복잡한 경우에만 작성한다.

- 테스트케이스ID
테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.
- 테스트 수행 절차
각 테스트를 수행하는데 있어 상세한 절차를 기술한다.

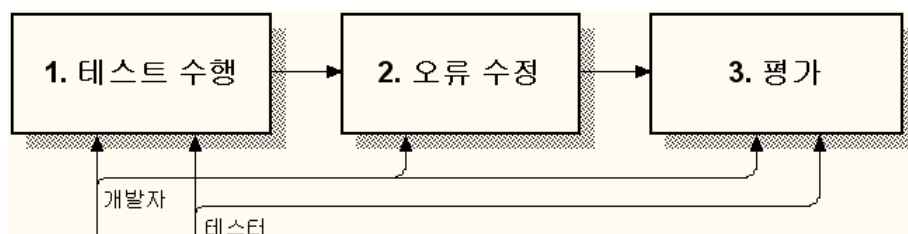
단위 테스트 수행 작업(3T22)

개요

단위 테스트 설계에 따라 컴포넌트, 사용자 인터페이스, 데이터베이스 프로그램 등의 테스트를 수행하고 오류를 수정하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 테스트 계획서 • 물리적 데이터베이스 • 컴포넌트 코드 • 사용자 인터페이스 코드 • 단위 테스트 기술서
산출물	<ul style="list-style-type: none"> • 단위 테스트 기술서
참여자	<ul style="list-style-type: none"> • 개발자 컴포넌트와 사용자 인터페이스, 데이터베이스 프로그램이 구현되면 단위 테스트 기술서의 테스트 절차에 따라 테스트를 실시하고 결과를 기록하며 오류가 발생되면 해당 코드를 수정한다. • 테스터 개발자를 대신하여 전문적인 테스트를 수행하고 결과를 기록할 수 있다. 발생한 오류사항을 개발자에게 전달한다.
관련기법	<ul style="list-style-type: none"> • 경험 기반 테스트 • 구조 기반 테스트 • 명세 기반 테스트

절차 흐름도



※ 오류 수정후 재 테스트를 수행해야 하고, 테스트 평가 결과 통과 기준을 충족하지 못했다면 통과할 때까지 절차 흐름을 계속 반복하면서 수행한다.

단위 테스트 수행 작업(3T22)

절차 설명

1. 테스트 수행

테스트케이스를 기반으로 단위 테스트를 수행한다.

- 단위 테스트 수행

테스트 계획서의 계획을 바탕으로 단위 테스트 기술서의 테스트케이스와 테스트 수행 절차에 따라서 테스트를 수행하고, 실제 결과나 오류 내용 등의 수행 결과를 단위 테스트 기술서의 테스트 결과에 기술한다.

2. 오류 수정

발견된 오류를 수정하고 재 테스트를 수행한다.

- 오류 수정

해당 테스트 결과 오류가 발생하였다면 이를 수정하고 재 테스트를 실시한다. 필요시 회귀 테스트(Regression Test)로써 오류와 연관된 다른 테스트케이스도 첫 번째 테스트에서 성공했을지라도 모두 다시 테스트할 수 있다. 재 테스트는 '1. 테스트 수행'을 따른다.

3. 평가

단위 테스트의 수행 결과를 평가해 테스트가 계획된 목표를 만족함을 보장한다.

- 평가

테스트 결과를 테스트 계획서에 명시된 테스트 특성별 측정 방법에 따라 평가하고 통과 기준을 충족하지 못하였다면 통과할 때까지 재 테스트 및 오류 수정을 반복한다. 필요시 상황에 맞도록 테스트 계획 작업의 테스트 특성 및 척도 정의 절차를 재수행한다.

작업 수행시 고려사항

■ 단위 테스트 수행 작업의 수행 여부

- 단위 테스트 수행 작업은 구현된 단위 소프트웨어를 테스트하는 작업으로써 반드시 실시해야 하나 일반적으로 컴포넌트 구현, 사용자 인터페이스 구현, 데이터베이스 구축 활동과 병행해서 실시할 수 있으므로 소규모 사업의 경우 별도로 분리해서 실시하지 않을 수 있다.

단위 테스트 수행 작업(3T22)

■ 테스트 주도 개발 적용시 수행 방법

- 구현 활동에서 테스트 주도 개발을 적용하였고 단위 테스트 활동과 구현 활동의 테스트케이스가 일치한다면 이로써 대체할 수 있다.

단위 테스트 수행 작업(3T22)

[서식] 단위 테스트 기술서(3T21a)

1.2 테스트 결과

1.2.1 (단위명)

테스트케이스 ID	실제 결과	테스트 수행일	테스트 수행자	오류 내용	조치내용	조치일	조치자

1.3 평 가

테스트 유형	테스트 유형 특성	측정 내용	측정 결과	평가 결과

단위 테스트 수행 작업(3T22)

■ [항목 설명] 단위 테스트 기술서(3T21a)

1.2 테스트 결과

1.2.1 (단위명)

테스트 단위의 명칭을 기재한다. 테스트 설계 항목과 동일하게 기재한다.

- 테스트케이스ID
테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.
- 실제 결과
테스트 입력 값을 처리한 후의 결과를 기재한다.
실제 결과가 예상 결과와 동일하면서 기재할 내용이 많으면 생략할 수 있다. 필요시 화면 캡처를 할 수 있다.
- 테스트 수행일
테스트를 수행한 일자를 기재한다.
- 테스트 수행자
테스트를 수행한 담당자명을 기재한다.
- 오류내용
오류 발생 시 오류의 내용에 대해 기재한다. 필요시 화면 캡처와 함께 기술할 수 있다.
- 조치내용
오류를 일으킨 테스트케이스가 어떻게 조치되었는지를 기술한다.
- 조치일
해당 오류가 조치된 일자를 기재한다.
- 조치자
해당 오류를 조치한 담당자명을 기재한다.

1.3 평가

- 테스트 유형
테스트 계획서에서 계획된 테스트 유형을 기재한다.
- 테스트 유형 특성
테스트 계획서에서 계획된 테스트 유형 특성을 기재한다.
- 측정 내용
테스트 계획서에서 계획된 측정 기준으로 측정된 내용을 기술한다.
- 측정 결과
테스트 계획서에서 계획된 측정 기준에 따른 측정 결과를 기재한다.
- 평가 결과
평가 결과로써 통과 여부를 기재한다.

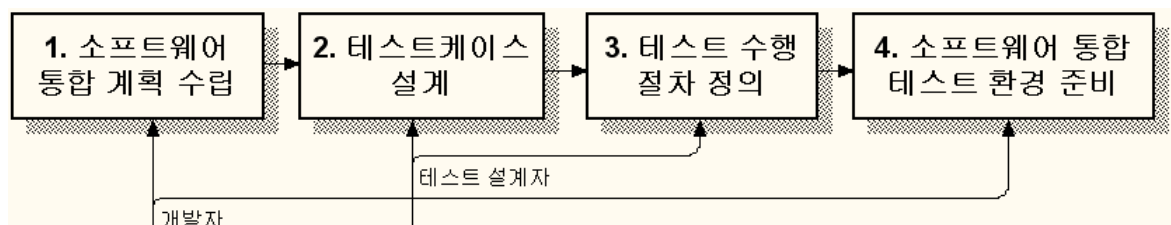
소프트웨어 통합 테스트 준비 작업(3T31)

개요

단위 테스트가 완료된 컴포넌트와 사용자 인터페이스, 데이터베이스를 통합하면서 통합 이상 여부 테스트를 설계하고 제반 사항을 준비하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 요구사항 정의서 유스케이스 정의서 컴포넌트 아키텍처 명세서 컴포넌트 명세서 컴포넌트 설계서 사용자 인터페이스 설계서 데이터베이스 설계서 연동 설계서 테스트 계획서 물리적 데이터베이스 사용자 인터페이스 코드 컴포넌트 코드
산출물	<ul style="list-style-type: none"> 소프트웨어 통합 테스트 기술서
참여자	<ul style="list-style-type: none"> 테스트 설계자 테스트 계획서를 바탕으로 소프트웨어 통합 테스트케이스와 수행 절차를 설계한다. 개발자 소프트웨어 통합 계획을 수립하고 테스트 수행을 위한 데이터를 개발한다.
관련기법	<ul style="list-style-type: none"> 경험 기반 테스트 구조 기반 테스트 명세 기반 테스트

절차 흐름도



소프트웨어 통합 테스트 준비 작업(3T31)

■ 절차 설명

1. 소프트웨어 통합 계획 수립

컴포넌트와 컴포넌트, 컴포넌트와 사용자 인터페이스 등을 점증적으로 통합하는 계획을 수립한다.

- 소프트웨어 통합 계획 수립

소프트웨어 단위를 통합하기 위한 계획을 수립한다. 단위 테스트를 완료한 소프트웨어 단위에 대해 각 단위를 통합하는 순서 및 절차를 작성하고 통합을 위한 전제조건 및 통합을 위한 소프트웨어, 하드웨어와 통합 테스트를 위한 테스트 스텝 등의 환경 요소를 도출한다.

2. 테스트케이스 설계

소프트웨어 통합에 따른 통합 테스트케이스를 설계한다.

- 테스트케이스 설계

테스트 계획서에서 정의된 테스트 유형에 따라 각 유형별 소프트웨어 통합 테스트를 위한 항목을 추출하고, 테스트 항목의 조건에 따라 분류되는 테스트케이스를 정의한다. 테스트케이스에 따라 관련 조건과 예상 결과(Test Oracle) 등을 기술한다.

3. 테스트 수행 절차 정의

각 테스트케이스 별로 입력에 따른 예상 출력을 얻기 위한 절차를 정의한다.

- 테스트 수행 절차 정의

테스트를 수행하는 순서 및 사전/제약조건을 정의한다.

4. 소프트웨어 통합 테스트 환경 준비

소프트웨어 통합 테스트 수행을 위한 테스트 환경을 준비한다.

- 테스트 프로그램 작성

컴포넌트를 통합하면서 컴포넌트의 외부 인터페이스 테스트를 위한 테스트 드라이버(Test Driver)나 테스트 스텝(Test Stub) 등이 필요할 경우 이를 제작한다. 테스트 드라이버란 테스트를 지원하기 위한 스크립트 코드나 데이터로써

소프트웨어 통합 테스트 준비 작업(3T31)

대상 컴포넌트의 동작을 호출한다. 테스트 스텝은 테스트 대상 컴포넌트가 호출하여 인터페이스할 더미(dummy) 컴포넌트에 해당한다.

- 데이터베이스 연결
데이터베이스와 연동되는 컴포넌트의 경우 데이터베이스 연결을 준비한다.
- 테스트 데이터 준비
입력값, 조회값 등 테스트케이스 정의에 따라 테스트에 활용될 데이터를 충분히 준비한다.

■ 작업 수행시 고려사항

■ 소프트웨어 통합의 종류 및 방법

- 동시식(big-bang) 통합
모든 모듈을 한꺼번에 통합하여 테스트를 수행한다. 통합 대상에서 중요 부분과 부수적인 부분을 구별하지 않고 오로지 목표 모습에 초점을 둔다.
테스트에 계획보다 많은 시간이 필요할 수 있으며 오류 발생시 원인 파악이 어렵다는 단점이 있다.
- 하향식(top-down) 통합
시스템 구조도의 위에 있는 모듈부터 아래층으로 내려오면서 통합을 진행한다. 상위층의 중요한 모듈 또는 골격을 먼저 테스트하게 된다.
점증적 통합이므로 오류의 원인을 찾기 쉽고, 테스트 스텝 제작 및 시스템의 작동 모습에 대한 파악이 용이하다.
- 상향식(bottom-up) 통합
최하위 모듈부터 통합하고 테스트를 진행한다.
점증적 통합이므로 오류의 원인을 찾기 쉬우나, 테스트에 계획보다 많은 시간이 필요할 수 있으며 초기에 시스템 골격이 갖춰지지 않아 시스템의 작동 모습 파악이 어렵다. 테스트 드라이버 제작이 필요하다.
- 연쇄식(threads) 통합
시스템의 중요한 기능을 담당하는 기본 기능 모듈부터 통합하고 테스트를 진행하는 가장 일반적인 방법이다. 기본 기능 모듈을 통합하면 이를 이용하는 모듈들을 연쇄적으로 통합하는 방법이다.
빠르게 통합할 수 있고 시스템의 작동 모습에 대한 파악이 용이하다
- 지속적(Continuous) 통합
지속적 통합(CI : Continuous Integration) 방법은 컴포넌트 구현 작업과 통합 작업을 분리하지 않고 구현과 동시에 통합시키는 기법이다.
자세한 내용은 기법서를 참고한다.

소프트웨어 통합 테스트 준비 작업(3T31)

■ 지속적 통합 적용시 수행 방법

- 지속적 통합을 활용할 경우에는 소프트웨어 통합 및 테스트 활동을 구현 활동과 동시에 실시할 수 있다.

■ 테스트케이스 설계시 고려사항

- 테스트케이스를 생성하는 방법에는 다음과 같은 접근 방법이 있다.
 - 테스트케이스 생성 도구 이용
 - 테스트 데이터 생성 도구 이용 또는 프로그램 개발
 - 유스케이스 시나리오로부터 생성
 - 각 설계서의 명세로부터 생성
- 테스트케이스는 다음 사항을 고려하여 설계한다.
 - 단위 테스트에서 성공한 테스트케이스일지라도 통합 후 문제가 발생할 수 있으므로 다른 소프트웨어 단위와 연결되는 테스트케이스라면 포함해야 한다.
 - 예외상황, 비정상적인 조건에서의 기능 테스트를 포함해야 한다.
 - 오류(Invalid) 데이터 입력 테스트를 포함해야 한다.
 - 계획된 테스트 커버리지(coverage)를 만족해야 한다.
- 소프트웨어 통합 테스트는 컴포넌트간의 통합과 서브 시스템간의 통합을 대상으로 테스트 항목과 테스트케이스를 정의한다.
- 컴포넌트에 대한 통합 테스트는 컴포넌트 아키텍처 명세서를 참조하여 각각의 컴포넌트로부터 호출관계가 있는 인터페이스를 식별하고, 그들이 속한 상호작용 다이어그램을 기반으로 각 컴포넌트 간에 메시지 전달을 검사할 수 있도록 테스트케이스를 개발한다.
- 서브시스템의 요구 기능이 제대로 동작하는 지에 대해서도 테스트케이스를 개발한다.

■ 테스트 수행 절차 정의시 고려사항

- 테스트케이스간 의존성이 있을 경우 이러한 사항을 사전조건으로 정의한다.
- 일반적 수행 절차 외에도 다음의 사항을 고려해야 한다.
 - 테스트를 몇 회 실시할 것인지
 - 어떤 조건으로 실시할 것인지
 - 어떻게 초기화할 것인지
 - 어떤 기술을 이용할 것인지

■ 소프트웨어 통합 테스트 준비 작업(3T31)

- 어떻게 측정할 것인지
- 결과를 기록하기 위한 특별한 방법이 있는지
- 어떤 조건이 되면 종료할 것인지 등이다.

■ 테스트 데이터 작성시 고려사항

- 테스트 데이터는 무작위로 생성된 garbage 데이터와 실제와 유사 또는 동등한 데이터로 나눌 수 있다.
- 일반적으로 실제와 유사한 데이터를 이용해서 테스트를 수행하지만 garbage 데이터를 단순 데이터 저장 및 전송 동작의 확인으로 충분한 경우, 볼륨 및 스트레스 테스트 수행, 오류 상황 연출 등의 목적을 위해 활용될 수 있다.

■ 상용 소프트웨어에 대한 테스트

- 만일 상용 소프트웨어(또는 컴포넌트)가 제안서의 규격에서 명시되지 않은, 응용 소프트웨어를 위한 세부적 기능이나 능력이 식별되면 응용 소프트웨어와 상용 소프트웨어를 통합하기 전에 이에 대한 테스트를 수행할 수 있다. (예 : DBMS의 트랜잭션 제어에 대한 특정 능력, 개발자 지원 편의 기능, 실행 환경 호환성 등)
- 상용 소프트웨어와 응용 소프트웨어의 통합성 테스트가 선결되어야 할 경우 상용 소프트웨어에 대한 테스트를 구현 및 테스트 단계 이전에 응용 소프트웨어 프로토타입을 이용하여 실시하는 것이 바람직하다.
- 만일 상용 소프트웨어를 분리발주 과정의 벤치마크 테스트를 거쳐 도입하였다면 이 과정은 별도로 수행하지 않는다.

■ 소프트웨어 통합 테스트 준비 작업(3T31)

■ [서식] 소프트웨어 통합 테스트 기술서(3T31a)

1. 테스트 설계

1.1. 테스트케이스

테스트 유형	테스트 항목	테스트 케이스ID	테스트 케이스명	관련 컴포넌트	입력값	사전조건	예상 결과

1.2. 테스트 수행 절차

테스트케이스ID	테스트 수행 절차

소프트웨어 통합 테스트 준비 작업(3T31)

■ [항목 설명] 소프트웨어 통합 테스트 기술서(3T31a)

1. 테스트 설계

1.1. 테스트케이스

- 테스트 유형

테스트 계획서의 테스트 유형을 참고하여 통합 테스트 대상에 해당하는 테스트 유형을 기재한다. 소프트웨어 통합 테스트에서는 컴포넌트 인터페이스 테스트, 컴포넌트 의존성 테스트, 서브시스템 의존성 테스트 등이 될 수 있다.

- 테스트 항목

테스트를 실제로 수행하는 단위를 기재한다.

- 테스트케이스ID

각 테스트케이스마다 유일하게 구분할 수 있는 식별자를 정의하여 기재한다.

- 테스트케이스명

각 테스트 항목에서 테스트 조건에 따라 분류되는 테스트케이스명을 정의하여 기재한다.

- 관련 컴포넌트

통합되는 컴포넌트나 비 컴포넌트명을 기재한다.

- 입력값

해당 테스트케이스를 위한 구체적인 입력 값을 기재한다.

- 사전조건

각 테스트케이스를 시작하기 전에 미리 갖추어야 할 조건이 있을 경우 이를 기술한다.

- 예상 결과

해당 테스트케이스를 수행할 때 입력 값에 의해 기대되는 결과를 기술한다.

1.2. 테스트 수행 절차

- 테스트케이스ID

테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.

- 테스트 수행 절차

각 테스트를 수행하는데 있어 상세한 절차를 기술한다.

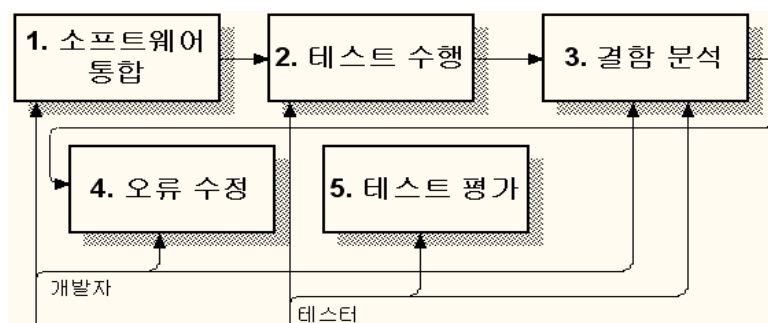
소프트웨어 통합 및 테스트 수행 작업(3T32)

개 요

소프트웨어 통합 테스트케이스와 테스트 수행 절차에 따라 테스트를 수행하고 오류를 수정하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 테스트 계획서 • 물리적 데이터베이스 • 컴포넌트 코드 • 사용자 인터페이스 코드 • 소프트웨어 통합 테스트 기술서
산출물	<ul style="list-style-type: none"> • 소프트웨어 통합 테스트 기술서
참여자	<ul style="list-style-type: none"> • 개발자 테스터가 테스트를 수행할 수 있도록 소프트웨어 통합 계획에 따라 컴포넌트들과 비 컴포넌트들을 통합하고, 오류가 발견된 코드를 분석 후 해당 오류를 수정한다. • 테스터 소프트웨어 통합 테스트를 수행하고, 발생한 오류 사항에 대해 개발자에게 전달하며 소프트웨어 통합 테스트 기술서의 테스트 결과와 평가를 작성한다.
관련기법	<ul style="list-style-type: none"> • 경험 기반 테스트 • 구조 기반 테스트 • 명세 기반 테스트 • 파레토 다이어그램 • 지속적 통합

절차 흐름도



※ 오류 수정후 재 테스트를 수행해야 하고, 테스트 평가 결과 통과 기준을 충족하지 못했다면 통과할 때까지 절차 흐름을 계속 반복하면서 수행한다.

소프트웨어 통합 및 테스트 수행 작업(3T32)

■ 절차 설명

1. 소프트웨어 통합

계획된 순서에 따라 소프트웨어 단위를 통합한다.

- 소프트웨어 통합
소프트웨어 통합 계획에 따라 컴포넌트와 비 컴포넌트를 통합한다.

2. 테스트 수행

소프트웨어 통합에 따른 테스트를 수행한다.

- 테스트 수행
테스트 계획서의 계획을 바탕으로 소프트웨어 통합 테스트 기술서의 테스트케이스와 테스트 수행 절차에 따라서 통합된 범위에 대한 테스트를 수행하고 실제 결과나 오류 내용 등의 수행 결과를 소프트웨어 통합 테스트 기술서의 테스트 결과에 기술한다. 통합된 범위에 대한 테스트가 완료되면 이와 또 다른 부분을 통합하고 테스트하는 과정을 반복한다.

3. 결함 분석

테스트 결과를 기반으로 결함 내역을 분석한다.

- 결함 분석
테스트가 종료되면 발생한 결함을 통합한다. 주원인별로 통합된 결함을 분석하고 빈번이 발생하는 결함 종류를 식별하기 위해 파레토 다이어그램과 같은 통계 기법을 사용한다.
- 결함 예방 방안 도출
결함에 대한 원인을 식별하고 이에 대한 예방 방안을 도출한다. 도출된 예방 방안을 개발자 및 관련자에게 회람하여 결함이 재발하지 않도록 한다.

4. 오류 수정

식별한 오류에 대해 수정하고 재 테스트를 실시한다.

- 오류 수정
테스트 결과 오류가 발생하였다면 이를 수정하고 재 테스트를 실시한다. 필요

■ 소프트웨어 통합 및 테스트 수행 작업(3T32)

시 회귀 테스트(Regression Test)로써 오류와 연관된 다른 테스트케이스도 첫 번째 테스트에서 성공했을지라도 모두 다시 테스트할 수 있다.

5. 테스트 평가

소프트웨어 통합 테스트 결과를 평가하고 소프트웨어 통합 테스트 활동을 종료한다.

- 테스트 평가 및 종료

테스트 결과를 테스트 계획서에 명시된 테스트 특성별 측정 방법에 따라 평가하고 통과 기준을 충족하지 못하였다면 통과할 때까지 재 테스트 및 오류 수정을 반복하고 필요시 상황에 맞도록 테스트 계획 작업의 테스트 특성 및 척도 정의 절차를 재수행한다.

■ 작업 수행시 고려사항

- 지속적 통합 적용시 수행 방법

- 지속적 통합을 활용할 경우에는 소프트웨어 통합 및 테스트 활동을 구현 활동과 동시에 실시할 수 있다.

■ 소프트웨어 통합 및 테스트 수행 작업(3T32)

■ [서식] 소프트웨어 통합 테스트 기술서(3T31a)

2. 테스트 결과

테스트케이스 ID	실제 결과	테스트 수행일	테스트 수행자	오류 내용	조치내용	조치일	조치자

3. 평 가

테스트 유형	테스트 유형 특성	측정 내용	측정 결과	평가 결과

소프트웨어 통합 및 테스트 수행 작업(3T32)

■ [항목 설명] 소프트웨어 통합 테스트 기술서(3T31a)

2. 테스트 결과

- 테스트케이스ID
테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.
- 실제 결과
테스트 입력 값을 처리한 후의 결과를 기술한다.
실제 결과가 예상 결과와 동일하면서 기술할 내용이 많으면 생략할 수 있다.
필요시 화면 캡처를 할 수 있다.
- 테스트 수행일
테스트를 수행한 일자를 기재한다.
- 테스트 수행자
테스트를 수행한 담당자명을 기재한다.
- 오류내용
오류 발생 시 오류의 내용에 대해 기술한다. 필요시 화면 캡처와 함께 기술할 수 있다.
- 조치내용
오류를 일으킨 테스트케이스가 어떻게 조치되었는지를 기술한다.
- 조치일
해당 오류가 조치된 일자를 기재한다.
- 조치자
해당 오류를 조치한 담당자명을 기재한다.

3. 평 가

- 테스트 유형
테스트 계획서에서 계획된 테스트 유형을 기재한다.
- 테스트 유형 특성
테스트 계획서에서 계획된 테스트 유형 특성을 기재한다.
- 측정 내용
테스트 계획서에서 계획된 측정 기준으로 측정된 내용을 기술한다.
- 측정 결과
테스트 계획서에서 계획된 측정 기준에 따른 측정 결과를 기재한다.
- 평가 결과
평가 결과로써 통과 여부를 기재한다.

시스템 통합 테스트 준비 작업(3T41)

개요

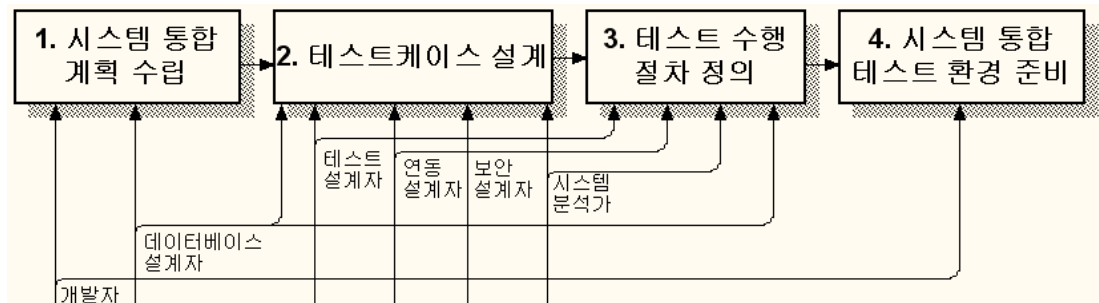
소프트웨어 통합 테스트가 완료된 소프트웨어와 하드웨어, 데이터, 네트워크를 통합하면서 통합 이상 여부 테스트를 설계하고 제반 사항을 준비하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 요구사항 정의서 • 시스템 아키텍처 정의서 • 유스케이스 정의서 • 컴포넌트 아키텍처 명세서 • 컴포넌트 설계서 • 사용자 인터페이스 설계서 • 데이터베이스 설계서 • 연동 설계서 • 자료 구축 설계서 • 보안 설계서 • 테스트 계획서 • 물리적 데이터베이스 • 컴포넌트 코드 • 사용자 인터페이스 코드
산출물	<ul style="list-style-type: none"> • 시스템 통합 테스트 기술서
참여자	<ul style="list-style-type: none"> • 테스트 설계자 테스트 계획서를 바탕으로 시스템 통합 테스트케이스와 수행 절차를 설계한다. • 연동 설계자 테스트 설계자와 함께 연동 테스트의 테스트케이스를 설계한다. • 보안 설계자 테스트 설계자와 함께 보안 분야 테스트의 테스트케이스를 설계한다. • 데이터베이스 설계자 테스트 설계자와 함께 자료 구축 및 전환 내역에 대한 검증 테스트케이스를 설계한다. • 시스템 분석가 테스트 설계자와 함께 시스템 성능 테스트의 테스트케이스를 설계한다.

시스템 통합 테스트 준비 작업(3T41)

구 분	작업 구성 항목
	<ul style="list-style-type: none"> 개발자 시스템 통합 계획을 수립하고 테스트 수행을 위한 데이터를 개발한다.
관련기법	<ul style="list-style-type: none"> 경험 기반 테스트 명세 기반 테스트

절차 흐름도



절차 설명

1. 시스템 통합 계획 수립

소프트웨어와 하드웨어, 데이터베이스, 네트워크 등을 점증적으로 통합하는 계획을 수립한다.

- 시스템 통합 계획 수립
소프트웨어, 하드웨어, 데이터베이스, 네트워크 등 시스템의 구성요소를 통합하기 위한 계획을 수립한다. 각 구성요소를 통합하는 순서 및 절차를 작성하고 통합을 위한 전제조건 및 시스템 통합 및 시스템 통합 테스트를 위한 환경을 정의한다.

2. 테스트케이스 설계

시스템 통합에 따른 통합 테스트케이스를 설계한다.

- 테스트케이스 설계
테스트 계획서에서 정의된 테스트 유형에 따라 각 유형별 소프트웨어 통합 테스트를 위한 항목을 추출하고, 테스트 항목의 조건에 따라 분류되는 테스트케

■ 시스템 통합 테스트 준비 작업(3T41)

이스를 정의한다. 테스트케이스에 따라 관련 조건과 예상결과(Test Oracle) 등을 기술한다.

3. 테스트 수행 절차 정의

각 테스트케이스 별로 입력에 따른 예상 출력을 얻기 위한 절차를 정의한다.

- 테스트 수행 절차 정의
테스트를 수행하는 순서 및 사전/제약조건을 정의한다.

4. 시스템 통합 테스트 환경 준비

시스템 통합 테스트 수행을 위한 테스트 환경을 준비한다.

- 테스트 환경 구축
시스템 통합 테스트 환경은 시스템의 운용 환경 및 사용자의 사용 환경과 최대한 유사하게 구현해야 한다.
- 테스트 데이터 준비
입력값, 조회값 등 테스트케이스 정의에 따라 테스트에 활용될 데이터를 충분히 준비한다.

■ 작업 수행시 고려사항

■ 테스트케이스 설계시 고려사항

- 테스트케이스를 생성하는 방법에는 다음과 같은 접근 방법이 있다.
 - 테스트케이스 생성 도구 이용
 - 테스트 데이터 생성 도구 이용 또는 프로그램 개발
 - 유스케이스 시나리오로부터 생성
 - 각 설계서의 명세로부터 생성
- 테스트케이스 유형은 다음 내용과 같으며 각 프로젝트 및 시스템의 성격에 따라 고려한다.
- 자료 구축 결과에 대한 검증 테스트
자료 전환 및 구축 건수, 수치 합계, 전체 데이터 비교, 시스템 조작 등의 테스트케이스를 포함할 수 있다.

시스템 통합 테스트 준비 작업(3T41)

- 기능성 테스트

구현된 시스템의 기능이 분석 단계에서 정의된 사용자 요구사항을 만족하는지 확인하는 테스트이다. 보통 블랙박스 테스트 기법을 활용하는 이 테스트는 요구사항의 충족여부를 테스트하는 것으로 사용자 지침서에 기술 되어 있는 대로 기능이 작동되는가를 테스트한다. 만약 해당 시스템의 요구 기능과 부합되지 않을 때에는 사용할 수 없다. 따라서 기능 테스트는 시스템이 갖추어야 할 최소한의 조건이 된다.

- 성능 테스트

성능 테스트는 시스템의 효율성을 진단하는 것으로 물리적인 제한조건 (CPU 처리속도, 기억용량, 데이터 전송 속도, 단말기 대수 등)내에서 해당 시스템의 응답속도, 처리량, 처리속도 등 요구되는 성능을 발휘하는가를 테스트한다. 성능 테스트는 다음과 같이 분류된다.

- 볼륨 테스트

시스템이 처리할 수 있는 최대 볼륨 한계를 테스트하는 것으로서 시스템 가동시 처리하게 될 데이터의 최대 볼륨보다 많은 양의 데이터를 처리해 보도록 하여 처리 가능한지를 테스트한다. 여기서 볼륨은 파일의 크기와 트랜잭션의 수, 또는 다른 측정 수단으로 정의된다. 볼륨 테스트는 종종 파일이 깨지는 경우가 발생하기 때문에 필요 이상으로 과다하게 테스트하지 않도록 주의하여야 한다.

- 스트레스 테스트

이 테스트는 시스템이 특정한 시간 동안 아주 많은 입력 데이터를 처리하도록 하는 것으로 시스템이 자원을 할당하고 공유하는 능력을 테스트한다. 이것은 동일한 상황이 이후에 또 발생할 경우 시스템이 효율적으로 작동하도록 하기 위해 고의적으로 특정 시간 동안 시스템 자원을 소모하는 것이다. 스트레스 테스트는 특히 온라인 트랜잭션 시스템의 경우 필요하다.

- 기억용량 테스트(storage test)

대용량의 데이터를 시스템에 입력시키고 그 반응을 테스트하는 것으로 데이터의 양이나 파일 크기가 변화함에 따라 시스템의 반응에 어떠한 변화가 있는가를 관찰해 보아야 한다. 대용량 테스트는 많은 양의 컴퓨터 자원과 인력을 소모하게 되므로 테스트를 얼마나 오래 계속할 것인가를 미리 계획할 필요가 있다. 이 테스트를 통하여 개발자는 일상작업을 수행하는데 충분한 기억용량을 추정할 수 있다.

시스템 통합 테스트 준비 작업(3T41)

- 복구성 테스트(recovery test)

개발된 시스템이나 시스템 구성요소 또는 응용 프로그램이 하드웨어적 오류, 회로 오류, 정전, 프로그램 오류 등 외부의 비정상적인 정지 또는 고장으로부터 정상적으로 원하는 시간 이내에 복구되어 가동 될 수 있는지를 확인한다. 이 테스트의 목적은 가용성에 있으며, 대개 비정상적인 동작 상황을 가정하여 강제적으로 시스템, 시스템의 구성요소 및 응용프로그램을 다운시킨 이후에 정상적으로 기동되는지를 확인하며, 다운된 시점부터 정상 동작될 때까지의 기간이 요구사항을 만족하는지 확인한다.

- 신뢰성 테스트

모든 기능에 대해 시스템이나 시스템 구성요소 또는 응용 프로그램이 다운되지 않고 안정적으로 수행되는지를 확인하고, 트랜잭션이 진행 중에 시스템이 다운되더라도 데이터의 무결성이 유지되는지 확인한다.

- 보안성 테스트

권한이 없는 사용자가 데이터에 접근 혹은 수정하는 것을 방지하는데 필수적인 특성이다. 보안 테스트는 시스템의 보안 특성이 업무환경에 적절한가를 테스트하는 것이다. 이러한 보안 문제는 전산 시스템이 점점 더 개방화되고 있는 상황에서 더욱 그 중요성이 커지고 있다.

- 호환성 테스트

현 시스템을 다른 하드웨어와 시스템 소프트웨어 환경으로 전환하거나 현재의 시스템에 중요한 새로운 기능을 추가하는 경우에 호환성에 대한 테스트케이스를 작성한다. 새로운 시스템 환경으로 전환하면서 발생하는 변경사항은 시스템의 모든 요소에 광범위하게 영향을 미친다. 따라서 호환성 테스트는 변경사항으로 인해 발생할 결점에 대비해 시스템의 모든 부분을 테스트 한다. 이를 위해 새로운 시스템의 테스트 결과와 비교할 수 있는 현재 시스템의 테스트 결과를 가지고 있어야 한다.

- 네트워크 테스트(network test)

네트워크의 모든 구성요소에 도달가능한지 확인하는 연결성을 확인하고 특정 네트워크 구간의 트래픽 또는 특정 업무 시간대의 트래픽을 테스트하고 모니터링 함으로써 네트워크의 가용성을 확인하는 테스트로써 단독적으로 수행하기보다는 스트레스 테스트, 볼륨 테스트와 함께 실시하거나, 시스템 오픈 초기에 모니터링의 방법으로 사용된다.

■ 시스템 통합 테스트 준비 작업(3T41)

■ 테스트 수행 절차 정의시 고려사항

- 테스트케이스간 의존성이 있을 경우 이러한 사항을 사전조건으로 정의한다.
- 일반적 수행 절차 외에도 다음의 사항을 고려해야 한다.
 - 테스트를 몇 회 실시할 것인지
 - 어떤 조건으로 실시할 것인지
 - 어떻게 초기화할 것인지
 - 어떤 기술을 이용할 것인지
 - 어떻게 측정할 것인지
 - 결과를 기록하기 위한 특별한 방법이 있는지
 - 어떤 조건이 되면 종료할 것인지 등이다.

■ 시스템 통합 테스트 환경 준비시 고려사항

- 만일 테스트를 위한 하드웨어, 네트워크, 장비 등이 현재 운용중이어서 테스트에 제약을 받는다면 주요 운용 시간을 회피하거나 가상 환경, 더미(dummy) 환경, 동일한 별도 환경 등을 구축한다.
- 사용자의 사용 환경은 운영체제, 브라우저 등이 다양할 수 있으므로 사용 환경에 대한 규격이 있다면 이에 맞게 준비해야 한다. 개발자의 개발 환경은 특정 라이브러리 등이 기 설치되어 있을 수 있으므로 개발 환경에서 테스트를 지양하고 깨끗한 테스트 환경을 준비해야 한다.
- 테스트 데이터는 무작위로 생성된 garbage 데이터와 실제와 유사 또는 동등한 데이터로 나눌 수 있다. 일반적으로 실제와 유사한 데이터를 이용해서 테스트를 수행하지만 garbage 데이터를 단순 데이터 저장 및 전송 동작의 확인으로 충분한 경우, 부름 및 스트레스 테스트 수행, 오류 상황 연출 등의 목적을 위해 활용될 수 있다.
- 테스트 수행 전에 테스트케이스 설계 결과를 요구사항 추적표에 반영하여 모든 요구사항에 대해 테스트가 수행되는지 확인해야 한다.

■ 지속적 통합 적용시 수행 방법

- 지속적 통합을 활용할 경우에는 시스템 통합 및 테스트 활동을 구현 활동과 동시에 실시할 수 있다.

■ 시스템 통합 테스트 준비 작업(3T41)

■ [서식] 시스템 통합 테스트 기술서(3T41a)

1. 테스트 설계

1.1. 테스트케이스

테스트 유형	테스트 항목	테스트 케이스ID	테스트 케이스명	관련 컴포넌트	입력값	사전조건	예상 결과	요구사항 ID

1.2. 테스트 수행 절차

테스트케이스ID	테스트 수행 절차

■ 시스템 통합 테스트 준비 작업(3T41)

■ [항목 설명] 시스템 통합 테스트 기술서(3T41a)

1. 테스트 설계

1.1. 테스트케이스

- 테스트 유형
테스트 계획서의 테스트 유형을 참고하여 시스템 테스트 대상에 해당하는 테스트 유형을 기재한다. 시스템 테스트에서는 시스템 기능 테스트, 시스템 비기능 테스트 등이 될 수 있다.
- 테스트 항목
테스트를 실제로 수행하는 단위를 기재한다.
- 테스트케이스ID
각 테스트케이스마다 유일하게 구분할 수 있는 식별자를 정의하여 기재한다.
- 테스트케이스명
각 테스트 항목에서 테스트 조건에 따라 분류되는 테스트케이스명을 정의하여 기재한다.
- 관련 컴포넌트
해당 테스트케이스를 수행하는데 관련이 있는 컴포넌트나 비 컴포넌트명을 기재한다.
- 입력값
해당 테스트케이스를 위한 구체적인 입력 값을 기재한다.
- 사전조건
각 테스트케이스를 시작하기 전에 미리 갖추어야 할 조건이 있을 경우 이를 기술한다.
- 예상 결과
해당 테스트케이스를 수행할 때 입력 값에 의해 기대되는 결과를 기술한다.
- 요구사항ID
해당 테스트케이스가 근거하는 요구사항을 요구사항 정의서의 요구사항ID로 기재한다.

1.2. 테스트 수행 절차

- 테스트케이스ID
테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.
- 테스트 수행 절차
각 테스트를 수행하는데 있어 상세한 절차를 기술한다.

시스템 통합 및 테스트 수행 작업(3T42)

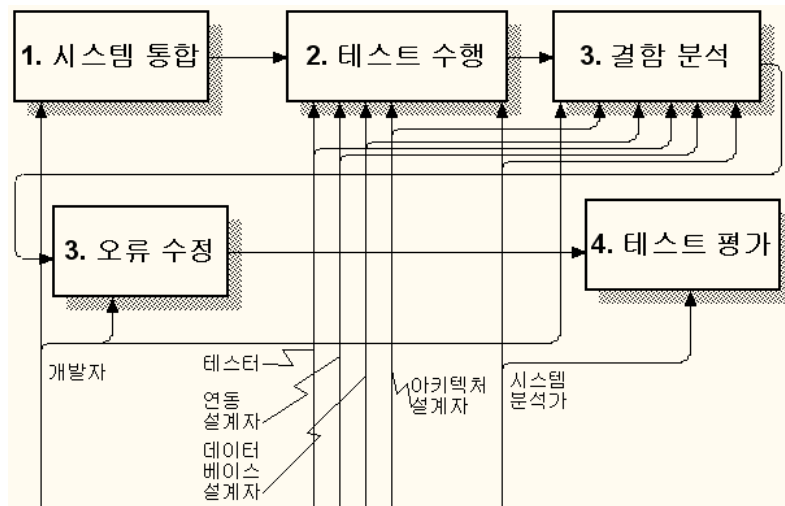
개요

시스템 통합 테스트케이스와 테스트 수행절차에 따라 테스트를 수행하고 오류를 수정하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 테스트 계획서 • 물리적 데이터베이스 • 컴포넌트 코드 • 사용자 인터페이스 코드 • 시스템 통합 테스트 기술서
산출물	<ul style="list-style-type: none"> • 시스템 통합 테스트 기술서
참여자	<ul style="list-style-type: none"> • 개발자 테스터가 테스트를 수행할 수 있도록 시스템 통합 계획에 따라 소프트웨어, 하드웨어, 데이터베이스, 네트워크 등을 통합하고, 오류가 발견된 코드를 분석 후 해당 오류를 수정한다. • 테스터 시스템 통합 테스트를 수행하고, 발생한 오류 사항에 대해 개발자에게 전달하며 시스템 통합 테스트 기술서의 테스트 결과를 작성한다. • 연동 설계자 테스터의 연동 시험을 점검하고 개발자와 함께 오류에 대한 분석을 실시한다. • 시스템 분석가 시스템 성능 시험 결과를 점검하고 시스템 테스트 기술서의 평가를 검토한다. • 데이터베이스 설계자 시스템 분석가와 함께 시스템 성능 시험 결과를 점검한다. • 아키텍처 설계자 시스템 분석가와 함께 시스템 성능 시험 결과를 점검한다.
관련기법	<ul style="list-style-type: none"> • 경험 기반 테스트 • 명세 기반 테스트 • 파레토 다이어그램 • 지속적 통합

시스템 통합 및 테스트 수행 작업(3T42)

절차 흐름도



※ 오류 수정후 재 테스트를 수행해야 하고, 테스트 평가 결과 통과 기준을 충족하지 못했다면 통과할 때까지 절차 흐름을 계속 반복하면서 수행한다.

절차 설명

1. 시스템 통합

계획된 순서에 따라 시스템 구성요소를 통합한다.

- 시스템 통합
시스템 통합 계획에 따라 소프트웨어와 하드웨어, 데이터베이스, 네트워크 등을 통합한다.

2. 테스트 수행

시스템 통합에 따른 테스트를 수행한다.

- 테스트 수행
테스트 계획서의 계획을 바탕으로 시스템 통합 테스트 기술서의 테스트케이스와 테스트 수행 절차에 따라서 테스트를 수행하고 실제 결과나 오류 내용 등의 수행 결과를 시스템 통합 테스트 기술서의 테스트 결과에 기술한다.

■ 시스템 통합 및 테스트 수행 작업(3T42)

3. 결함 분석

테스트 결과를 기반으로 결함 내역을 분석한다.

- 결함 분석

테스트가 종료되면 발생한 결함을 통합한다. 주원인별로 통합된 결함을 분석하고 빈번이 발생하는 결함 종류를 식별하기 위해 파레토 다이어그램과 같은 통계 기법을 사용한다.

- 결함 예방 방안 도출

결함에 대한 원인을 식별하고 이에 대한 예방 방안을 도출한다. 도출된 예방 방안을 개발자 및 관련자에게 회람하여 결함이 재발하지 않도록 한다.

4. 오류 수정

식별한 오류에 대해 수정하고 재 테스트를 실시한다.

- 오류 수정

해당 테스트 결과 오류가 발생하였다면 이를 수정하고 재 테스트를 실시한다. 필요시 회귀 테스트(Regression Test)로써 오류와 연관된 다른 테스트케이스도 첫 번째 테스트에서 성공했을지라도 모두 다시 테스트할 수 있다.

5. 테스트 평가

시스템 통합 테스트 결과를 평가하고 시스템 통합 테스트 활동을 종료한다.

- 테스트 평가 및 종료

테스트 결과를 테스트 계획서에 명시된 테스트 특성별 측정 방법에 따라 평가하고 통과 기준을 충족하지 못하였다면 통과할 때까지 재 테스트 및 오류 수정을 반복하고 필요시 상황에 맞도록 테스트 계획 작업의 테스트 특성 및 척도 정의 절차를 재수행한다.

■ 작업 수행시 고려사항

■ 지속적 통합 적용시 수행 방법

- 지속적 통합을 활용할 경우에는 시스템 통합 및 테스트 활동을 구현 활동과 동시에 실시할 수 있다.

■ 시스템 통합 및 테스트 수행 작업(3T42)

■ [서식] 시스템 통합 테스트 기술서(3T41a)

2. 테스트 결과

테스트케이스 ID	실제 결과	테스트 수행일	테스트 수행자	오류 내용	조치내용	조치일	조치자

3. 평 가

테스트 유형	테스트 유형 특성	측정 내용	측정 결과	평가 결과

시스템 통합 및 테스트 수행 작업(3T42)

■ [항목 설명] 시스템 통합 테스트 기술서(3T41a)

2. 테스트 결과

- 테스트케이스ID
테스트케이스 항목의 테스트케이스 ID와 동일하게 기재한다.
- 실제 결과
테스트 입력 값을 처리한 후의 결과를 기술한다.
실제 결과가 예상 결과와 동일하면서 기술할 내용이 많으면 생략할 수 있다.
필요시 화면 캡처를 할 수 있다.
- 테스트 수행일
테스트를 수행한 일자를 기재한다.
- 테스트 수행자
테스트를 수행한 담당자명을 기재한다.
- 오류내용
오류 발생 시 오류의 내용에 대해 기술한다. 필요시 화면 캡처와 함께 기술할 수 있다.
- 조치내용
오류를 일으킨 테스트케이스가 어떻게 조치되었는지를 기술한다.
- 조치일
해당 오류가 조치된 일자를 기재한다.
- 조치자
해당 오류를 조치한 담당자명을 기재한다.

3. 평 가

- 테스트 유형
테스트 계획서에서 계획된 테스트 유형을 기재한다.
- 테스트 유형 특성
테스트 계획서에서 계획된 테스트 유형 특성을 기재한다.
- 측정 내용
테스트 계획서에서 계획된 측정 기준으로 측정된 내용을 기술한다.
- 측정 결과
테스트 계획서에서 계획된 측정 기준에 따른 측정 결과를 기재한다.
- 평가 결과
평가 결과로써 통과 여부를 기재한다.

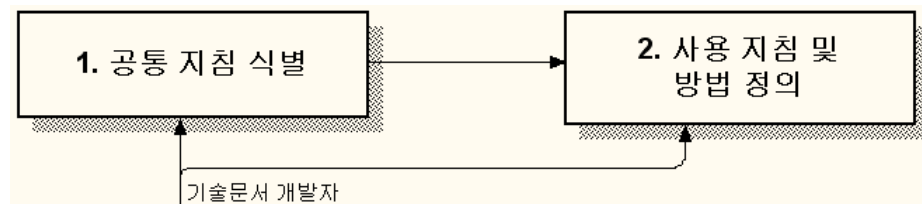
■ 사용자 지침서 작성 작업(3T51)

■ 개 요

사용자가 개발된 시스템을 잘 이해하고 기능을 올바르게 사용할 수 있도록 사용 방법과 유의사항을 기술한 문서를 작성하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 용어집 • 유스케이스 정의서 • 사용자 인터페이스 명세서 • 사용자 인터페이스 설계서
산출물	<ul style="list-style-type: none"> • 사용자 지침서
참여자	<ul style="list-style-type: none"> • 기술문서 개발자 <p>시스템에 대하여 사용자 지침서 등 최종 사용자를 지원하는 자료들을 개발한다.</p>
관련기법	해당 사항 없음

■ 절차 흐름도



■ 절차 설명

1. 공통 지침 식별

시스템 전반에 걸친 공통 사용 지침을 식별해 정의한다.

• 공통 지침 식별

사용자가 지침서에서 원하는 항목을 쉽게 찾아볼 수 있고 중복된 기술을 제거하기 위해 공통으로 기술하여야 하는 항목을 식별한다. 예를 들어 사용자 지침서를 작성하는 목적 및 시스템 사용 시 필요한 설치 방법, 시작 및 종료 방법, 시스템에서 제공하는 모든 기능의 공통적인 사항에 대한 사용 지침 등이 있다.

■ 사용자 지침서 작성 작업(3T51)

2. 사용 지침 및 방법 정의

화면별로 사용 지침 및 방법을 정의한다.

- 사용 지침 및 방법 정의

실제 화면을 제시하면서 기능별로 사용법을 상세하게 기술하고 각 기능별 주의 사항과 발생 가능한 오류 메시지, 오류의 원인, 오류의 해결 방법 등을 기술한다.

■ 작업 수행시 고려사항

■ 사업 특성에 맞도록 서식 보완

- 사용자 지침서는 궁극적으로 사용자가 시스템을 잘 이해하고 올바른 기능 조작을 유도하기 위해서 작성된다. 따라서 사용자의 수준을 고려하고 사용자의 요구사항에 따라 서식은 보완되어야만 한다. 또한 시스템을 사용하는 조직의 지침(예 : 업무지침 또는 보안지침 등), 개발 시스템의 유형(예 : 이동식 장비를 위한 핸드북 등) 등의 요소도 고려되어야 한다.
- 특히 사용자 지침서와 교육 자료, 온라인 도움말 등이 각각 작성되어 자료 중복으로 인한 혼란, 갱신후 각 내용 불일치 등의 문제가 발생되지 않도록 가급적 통합된 용도의 지침서가 작성되도록 해야 한다.
- 일반 사용자뿐만 아니라 관리자의 사용 지침 및 방법도 기술해야 하며 필요시 지침서를 분권할 수 있다.

■ 사용자 지침서 작성 작업(3T51)

■ [서식] 사용자 지침서(3T51a)

1. 개요

2. 사용 지침

2.1. 설치 방법

2.2. 시작 및 종료 방법

2.3. 공통 사용 지침

3. 기능별 사용 방법

3.1. (기능명)

3.1.1. (화면명(화면ID))

3.1.1.1. 비즈니스 흐름

3.1.1.2. 사용 방법

3.1.1.3. 주의 사항

3.1.1.4. 오류 메시지 및 처리 방법

오류 메시지	원인	처리 방법

■ 사용자 지침서 작성 작업(3T51)

■ [항목 설명] 사용자 지침서(3T51a)

1. 개요

사용자 지침서를 작성하는 목적 및 사용 대상자, 시스템 특징 등의 내용을 상세하게 기술한다.

2. 사용 지침

2.1. 설치 방법

사용자 환경에 시스템의 설치 방법을 기술한다.

2.2. 시작 및 종료 방법

사용자 환경의 시스템 시작 및 종료 방법을 기술한다.

2.3. 공통 사용 지침

시스템에서 제공하는 모든 기능의 공통적인 사항들 즉, 화면 구성 및 다루기, 메뉴 구성 및 다루기, 아이콘, 기능키에 대한 설명을 기술한다.

3. 기능별 사용 방법

3.1. (기능명)

기능명 및 기능ID를 기재하고 기능에 대한 간략한 설명을 기술한다.

3.1.1. (화면명(화면ID))

화면명 및 화면ID를 기재한다.

3.1.1.1. 비즈니스 흐름

업무 프로세스에서 해당 화면의 역할, 이 화면을 조작하기 위한 선행/후행 화면과의 관계 등을 서술하고 적절한 다이어그램을 조합하여 기술할 수 있다.

3.1.1.2. 사용 방법

실제 화면 모습을 제시하면서 시스템이 갖는 기능과 사용 방법에 대한 설명을 자세히 기술한다.

3.1.1.3. 주의 사항

해당 화면을 조작하면서 사용자가 주의해야 하는 사항을 기술한다.

■ 사용자 지침서 작성 작업(3T51)

3.1.2. 오류 메시지 및 처리 방법

발생 가능한 오류와 그 원인 및 처리 절차에 대하여 기술한다.

- 오류 메시지
오류 상황이 되었을 경우 표시되는 메시지의 내용을 기재한다.
- 원인
오류 상황이 발생한 원인을 기술한다.
- 처리 방법
오류 상황에 대한 처리 방법을 기술하되, 사용자 입장에서 해야 하는 방법을 알기 쉽게 기술한다.

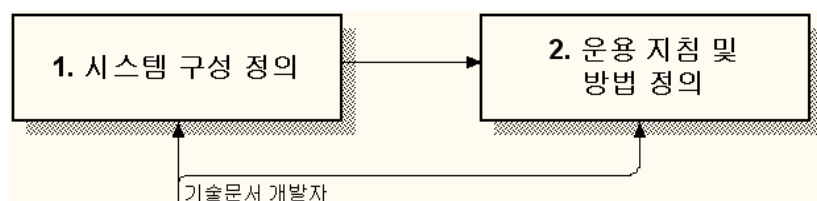
■ 운용자 지침서 작성 작업(3T52)

■ 개 요

운용자가 개발된 시스템의 운용 방법을 정확히 이해하고 운용하며, 장애 상황시 신속하고 적절하게 조치할 수 있도록 방법과 주의 사항을 기술한 문서를 작성하는 작업이다.

구 분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 시스템 아키텍처 정의서 • 연동 소요 정의서 • 데이터베이스 설계서 • 연동 설계서 • 시스템 보안 설계서 • 시스템 설치 계획서
산출물	<ul style="list-style-type: none"> • 운용자 지침서
참여자	<ul style="list-style-type: none"> • 기술문서 개발자 <p>시스템에 대하여 운용자 지침서 등 최종 사용자를 지원하는 자료들을 개발한다.</p>
관련기법	해당 사항 없음

■ 절차 흐름도



■ 절차 설명

1. 시스템 구성 정의

시스템 운용에 필요한 하드웨어, 네트워크 등 시스템의 기반 환경 및 소프트웨어의 구성을 정의한다.

- 시스템 구성 정의
시스템 아키텍처 정의서와 기타 문서를 참조하여 시스템을 구성하는 소프트웨어, 하드웨어, 네트워크 등을 명시하고 설명할 수 있는 내용을 정의한다.

운용자 지침서 작성 작업(3T52)

2. 운용 지침 및 방법 정의

실제 운용 환경을 기반으로 운용 지침 및 방법을 정의한다.

- **운용 지침 및 방법 정의**

운용자가 시스템을 정확히 이해하고 운용할 수 있도록 실제 화면을 제시하면서 설치, 시작 및 종료, 백업, 정기 작업, 장애 복구 방법 등을 상세하게 기술하고 각 기능별 주의 사항과 발생 가능한 오류 메시지, 오류의 원인, 오류의 해결 방법 등을 기술한다.

작업 수행시 고려사항

- **사업 특성에 맞도록 서식 보완**

- 운용자 지침서는 궁극적으로 운용자가 시스템을 정확히 이해하고 장애시 신속하고 적절한 조치가 가능하게 하기 위해서 작성된다. 따라서 시스템을 운용하는 곳의 지침, 개발 시스템의 유형 등의 요소에 따라 서식이 달라져야 한다.

■ 운용자 지침서 작성 작업(3T52)

■ [서식] 운용자 지침서(3T52a)

1. 개요

2. 시스템 구성

2.1. 시스템 구성도

2.2. 운용 환경 구성

장 소	설 명	설치 하드웨어

2.3. 하드웨어 구성

하드웨어명	사 양

2.4. 소프트웨어 구성

하드웨어명	유형	소프트웨어명	설명	제조사 /납품사

2.5. 네트워크 구성

통신장비명	설명	수량	모델명	제조사 /납품사	연결 하드웨어	설치 장소

■ 운용자 지침서 작성 작업(3T52)

3. 시스템 설치 절차

3.1. 네트워크 통신장비

3.2. 하드웨어

3.3. 패키지/시스템 소프트웨어

3.4. 응용 소프트웨어

4. 시스템 운용

4.1. 시작 및 종료

4.2. 자료 백업

4.3. 정기 작업

4.4. 장애 복구

4.5. 보안 유의사항

5. 기타사항

부록.

■ **운용자 지침서 작성 작업(3T52)**

■ **[항목 설명] 운용자 지침서(3T52a)**

1. 개요

운용자 지침서를 작성하는 목적 및 운용 대상자, 시스템 특징 등의 내용을 상세하게 기술한다.

2. 시스템 구성

2.1. 시스템 구성도

운용될 시스템의 분산 환경과 구성 요소 등의 관계 및 동작 방식을 한눈에 파악할 수 있는 구성도를 작성한다.

2.2. 운용 환경 구성

운용 환경은 시스템의 운용 장소가 통합 또는 여러 장소에 분산되어 있는지에 대한 기술을 한다.

- 장소
시스템이 운용되는 실질적 장소(노드)를 기재한다. (예 : 00부대 전산실)
- 설명
장소에 대한 간략한 설명을 기술한다.
- 설치 하드웨어
장소에 설치된 하드웨어명을 기재한다.

2.3. 하드웨어 구성

- 하드웨어명
시스템을 구성하는 하드웨어명을 기재한다.
- 사양
각 하드웨어의 사양(제조사/납품사, 모델명, CPU, 메모리, Disk 용량, Port수, OS 등)에 대해 자세히 기술한다.

2.4. 소프트웨어 구성

- 하드웨어명
소프트웨어가 탑재되는 하드웨어명을 기재한다.
- 유형
소프트웨어의 유형(예 : WAS, DBMS 등)을 기재한다.
- 소프트웨어명
소프트웨어명과 버전을 기재한다.

■ **운용자 지침서 작성 작업(3T52)**

- 설명
시스템에서 소프트웨어의 역할에 대해 간략히 기술한다.
- 제조사/납품사
소프트웨어를 제조한 제조사와 기술 지원을 담당한 납품사명을 기재한다.

2.5. 네트워크 구성

아래 내용을 포함하여 네트워크 구성 방식 및 특징을 기술하고 필요시 네트워크 구성도를 작성한다.

- 통신장비명
네트워크를 구성하는 통신장비명을 기재한다.
- 설명
통신장비에 대한 간략한 설명을 기술한다.
- 수량
통신장비 수량을 기재한다.
- 모델명
통신장비의 제조 모델명을 기재한다.
- 제조사/납품사
통신장비에 대한 제조사와 기술 지원을 담당한 납품사명을 기재한다.
- 연결 하드웨어
통신장비와 통신하는 하드웨어명을 기재한다.
- 설치 장소
통신장비가 운용되는 실질적 노드(장소)명을 기재한다. (예 : 00부대 전산실)

3. 시스템 설치 절차

시스템 설치 절차는 납품자 입장에서의 설치가 아닌, 운용자 입장에서 시스템 초기화 설치를 위한 최소한의 정보만을 선별해서 기술한다. 만일 별도의 매뉴얼이 존재한다면 이에 대한 참조 언급을 할 수 있다.

3.1. 네트워크 통신장비

시스템의 정상적 운용을 위한 네트워크 통신장비의 세팅 절차를 기술한다.

3.2. 하드웨어

시스템의 정상적 운용을 위한 하드웨어의 세팅 절차를 기술한다.

■ 운용자 지침서 작성 작업(3T52)

3.3. 패키지/시스템 소프트웨어

시스템의 정상적 운용을 위한 패키지/시스템 소프트웨어의 세팅 절차를 기술한다.

3.4. 응용 소프트웨어

시스템의 정상적 운용을 위한 응용 소프트웨어의 세팅 절차를 기술한다.

4. 시스템 운용

4.1. 시작 및 종료

운용자 입장에서 시스템의 시작 및 종료 방법을 기술한다.

4.2. 자료 백업

자료의 운용 및 백업 방법을 기술한다.

4.3. 정기 작업

정상적인 시스템 운용을 위해 운용자가 정기적으로 수행해야 할 작업(확인, 점검, 정비 등)의 내용을 기술한다.

4.4. 장애 복구

시스템 운용 중 발생할 수 있는 장애를 열거하고 조치방법을 기술한다.

4.5. 보안 유의사항

시스템 운용자가 숙지하고 있어야 할 보안 유의사항에 대해 기술한다.

5. 기타사항

납품사 또는 유지보수 전담 기술팀 연락처, 유지보수팀 연락처 등 운용자가 참고해야 할 내용을 기술한다.

부록.

별도의 매뉴얼이나 프로그램(예 : 웹 컴포넌트, DBMS 프로그램) 리스트 등을 부록 또는 별권으로 첨부한다.

인도 단계(4S)

개요

인도 단계는 개발된 시스템을 실제 운용 환경에 설치하고 인수 테스트 및 교육 등을 지원하는 것을 목적으로 한다. 인도 단계는 시스템 설치, 인수지원 활동으로 구성된다.

활동/작업 설명

1. 시스템 설치(4S1)

시스템 설치 활동은 기 수립된 시스템 설치 계획에 따라 지정된 장소에 플랫폼을 설치하고 기존 시스템의 데이터를 전환하며 개발된 응용 및 상용 소프트웨어를 설치하여 시스템이 운용될 수 있는 상태로 준비한다.

작업ID	작업명	작업내용
4S11	시스템 설치 실시	개발된 시스템을 운용할 수 있도록 시스템 설치 계획서 등을 바탕으로 시스템을 운용 환경에 설치한 후 이상 여부에 대해 테스트를 실시한다.

2. 인수 지원(4S2)

인수 지원은 사용자 주관의 시험평가를 지원하고 시스템에 대한 교육을 실시하여 사용자 및 운용자가 시스템을 사용할 수 있는 역량을 갖추도록 한다.

작업ID	작업명	작업내용
4S21	인수 지원 실시	사용자 요구사항을 모두 충족하는 시스템이 개발되었는지 확인하기 위한 시험평가 수행을 지원한다. 또한 신규 시스템을 운용하고 사용할 시스템 운용자와 사용자, 관리자 등을 대상으로 시스템 운용 방법과 사용 방법 등을 교육한다.

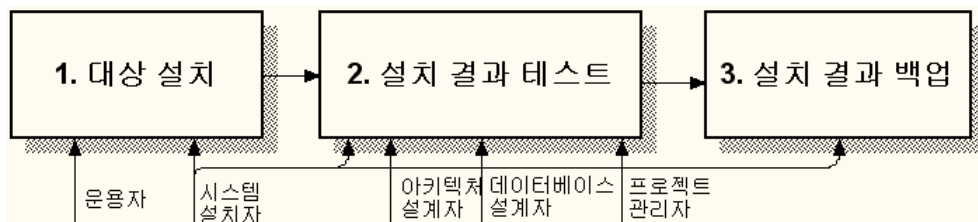
시스템 설치 실시 작업(4S11)

개요

개발 대상 시스템이 구현 완료되면 발주자가 지정한 장소와 플랫폼에 납품자가 시스템 설치 계획에 따라 시스템을 설치하고 결과 테스트를 수행하는 작업이다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> • 시스템 아키텍처 정의서 • 시스템 설치 계획서 • 물리적 데이터베이스 • 사용자 인터페이스 코드 • 컴포넌트 코드
산출물	<ul style="list-style-type: none"> • 시스템 설치 결과서
참여자	<ul style="list-style-type: none"> • 프로젝트 관리자 시스템 설치 계획을 바탕으로 시스템 설치 과정을 점검하고 시스템 설치 결과서를 작성한다. • 시스템 설치자 하드웨어, 네트워크, 소프트웨어, 개발된 응용 시스템을 설치하고 이상 여부에 대한 테스트를 실시한다. • 아키텍처 설계자 설치 대상 시스템이 시스템 아키텍처에 부합하는지에 대한 점검을 수행한다. • 데이터베이스 설계자 데이터베이스를 구축하고 이상 여부에 대한 테스트를 실시한다. • 운용자 설치 과정에 따른 행정절차 및 필요한 제반 도구에 대해 협조한다.
관련기법	해당 사항 없음

절차 흐름도



■ 시스템 설치 실시 작업(4S11)

■ 절차 설명

1. 대상 설치

시스템 설치 계획에 따라 하드웨어, 네트워크 및 소프트웨어를 설치한다.

- 운용 환경 구성

시스템 설치 계획서에 따라 네트워크 통신장비, 하드웨어, 패키지/시스템 소프트웨어를 운용 환경에 설치한다. 만일 기존 시스템이 존재한다면 시스템 설치 계획에 따라 기존 시스템을 백업하고 전환한다.

- 응용 소프트웨어 설치 및 자료 구축

응용 소프트웨어, 데이터 등을 개발 환경에서 실제 운용 환경으로 이관하여 설치한다.

2. 설치 결과 테스트

시스템 설치 결과를 검증한다.

- 설치 결과 테스트

각 설치 대상이 이상 없이 설치되었는가를 확인하기 위해 시스템 설치 계획서에 따라 테스트를 수행한다. 테스트 성공 후 정확성 검증을 위해 생성된 임시 파일, 임시 데이터, 임시 로그, 임시 권한 등은 모두 삭제한다.

3. 설치 결과 백업

운용 시 활용할 수 있도록 시스템 설치 결과를 백업한다.

- 설치 결과 백업

테스트 성공 후 신규 시스템이 자동으로 세팅되지 않고 수동으로 세팅되어야 하는 요소가 있다면 시스템 설치 후 파라미터 세팅 파일 등 신규 시스템의 초기 상태를 백업한다.

■ 작업 수행시 고려사항

■ 시스템 설치 실시 작업의 수행 여부

- 시스템 설치 실시 작업은 소프트웨어만을 납품하는 등 해당 사업이 시스템 설치에 대한 과업이 포함되어 있지 않다면 생략될 수 있다.

■ 시스템 설치 실시 작업(4S11)

■ 설치 결과 테스트 중 유의사항

- 시스템 설치 후 테스트 시 시스템에 오류가 발생하면 수정 작업을 한다. 이때 수정 작업이 예상보다 많은 자원(시간, 비용)이 소요될 경우에는 개발된 시스템의 오류 중요도를 검토하여 계속 작업을 진행할 것인지 작업을 중지하고 기존 시스템으로 복귀시키고 추후 계획을 수립하여 다시 설치 작업을 수행할 것 인지를 결정하도록 한다.

■ 작업 시점의 조정

- 소프트웨어 개발 공정과 하드웨어 등의 기반 요소 납품 및 설치 시점이 일치하지 않을 수 있으므로 모든 설치 대상이 후반부에 한꺼번에 설치되어야 할 필요가 없고 사업에 특성에 맞게 계획을 수립한 후 계획에 맞게 설치될 수 있다.

시스템 설치 실시 작업(4S11)

[서식] 시스템 설치 결과서(4S11a)

1. 설치 작업 결과

1.1. 네트워크 통신장비

1.1.1. (이름)

이름	설치일		담당자	
설치 내용	수량	설치 장소	네트워크	LAN
설치 및 테스트 결과				
특이사항				

1.2. 하드웨어

1.2.1. (이름)

이름	설치일		담당자	
설치 내용	수량	설치 장소	네트워크	LAN
설치 및 테스트 결과				
특이사항				

1.3. 패키지/시스템 소프트웨어

1.3.1. (이름)

이름	설치일		담당자	
설치 내용	수량		하드웨어	
설치 및 테스트 결과				
특이사항				

시스템 설치 실시 작업(4S11)

1.4. 응용 소프트웨어

1.4.1. (이름)

이름	설치일		담당자
설치 내용	하드웨어	하드웨어 경로	배포 파일명
설치 및 테스트 결과			
특이사항			

1.5. 구축 데이터

1.5.1. (이름)

이름			설치일		담당자	
설치 내용	DBMS	테이블ID				
설치 및 테스트 결과						
특이사항						

시스템 설치 실시 작업(4S11)

■ [항목 설명] 시스템 설치 결과서(4S11a)

1. 설치 작업 결과

설치 대상을 구분하고 식별자 및 이름을 목록으로 작성한다.

1.1. 네트워크 통신장비

1.1.1. (이름)

- 이름
통신장비를 대표하는 명칭을 기재하되, 시스템 설치 계획서에 기재된 명칭을 기재한다.
- 설치일
설치 완료 날짜를 기재한다.
- 담당자
설치를 담당한 담당자를 기재한다.
- 수량
설치한 통신장비의 수량을 기재한다.
- 설치 장소
통신장비를 설치한 장소를 기재하되 시스템 노드명으로 기재한다.
- 네트워크
통신장비가 연결된 네트워크 명칭을 기재한다.
- LAN
통신장비가 연결된 LAN 명칭을 기재한다.
- 설치 및 테스트 결과
설치 및 테스트의 완료 여부를 간략히 기술한다.
- 특이사항
설치 과정에서 식별된 문제점이나 향후 보완 사항을 기술한다.

1.2. 하드웨어

1.2.1. (이름)

- 이름
하드웨어를 대표하는 명칭을 기재하되, 시스템 설치 계획서에 기재된 명칭을 기재한다.
- 설치일
설치 완료 날짜를 기재한다.
- 담당자
설치를 담당한 담당자를 기재한다.

시스템 설치 실시 작업(4S11)

- 수량
설치한 하드웨어의 수량을 기재한다.
- 설치 장소
하드웨어를 설치한 장소를 기재하되 시스템 노드명으로 기재한다.
- 네트워크
하드웨어가 연결된 네트워크 명칭을 기재한다.
- LAN
하드웨어가 연결된 LAN 명칭을 기재한다.
- 통신장비
하드웨어가 연결된 통신장비의 명칭을 기재한다.
- 설치 및 테스트 결과
설치 및 테스트의 완료 여부를 간략히 기술한다.
- 특이사항
설치 과정에서 식별된 문제점이나 향후 보완 사항을 기술한다.

1.3. 패키지/시스템 소프트웨어

1.3.1. (이름)

- 이름
패키지/시스템 소프트웨어를 대표하는 명칭을 기재하되, 시스템 설치 계획서에 기재된 명칭을 기재한다.
- 설치일
설치 완료 날짜를 기재한다.
- 담당자
설치를 담당한 담당자를 기재한다.
- 수량
설치한 패키지/시스템 소프트웨어의 수량을 기재한다.
- 하드웨어
패키지/시스템 소프트웨어가 탑재된 하드웨어 명칭을 기재한다.
- 설치 및 테스트 결과
설치 및 테스트의 완료 여부를 간략히 기술한다.
- 특이사항
설치 과정에서 식별된 문제점이나 향후 보완 사항을 기술한다.

시스템 설치 실시 작업(4S11)

1.4. 응용 소프트웨어

1.4.1. (이름)

- 이름
응용 소프트웨어를 대표하는 명칭을 기재한다.
- 설치일
설치 완료 날짜를 기재한다.
- 담당자
설치를 담당한 담당자를 기재한다.
- 하드웨어
응용 소프트웨어가 탑재된 하드웨어 명칭을 기재한다.
- 하드웨어 경로
응용 소프트웨어가 탑재된 하드웨어의 디렉토리 경로를 기재한다.
- 배포 파일명
응용 소프트웨어가 하드웨어에 탑재(배포)된 파일명을 기재한다.
- 설치 및 테스트 결과
설치 및 테스트의 완료 여부를 간략히 기술한다.
- 특이사항
설치 과정에서 식별된 문제점이나 향후 보완 사항을 기술한다.

1.5. 구축 데이터

1.5.1. (이름)

- 이름
구축 데이터를 대표하는 명칭을 기재한다.
- 설치일
설치 완료 날짜를 기재한다.
- 담당자
설치를 담당한 담당자를 기재한다.
- DBMS
구축 데이터가 탑재된 DBMS를 기재한다.
- 테이블ID
구축 데이터가 삽입된 테이블ID를 기재한다.
- 설치 및 테스트 결과
설치 및 테스트의 완료 여부를 간략히 기술한다.
- 특이사항
설치 과정에서 식별된 문제점이나 향후 보완 사항을 기술한다.

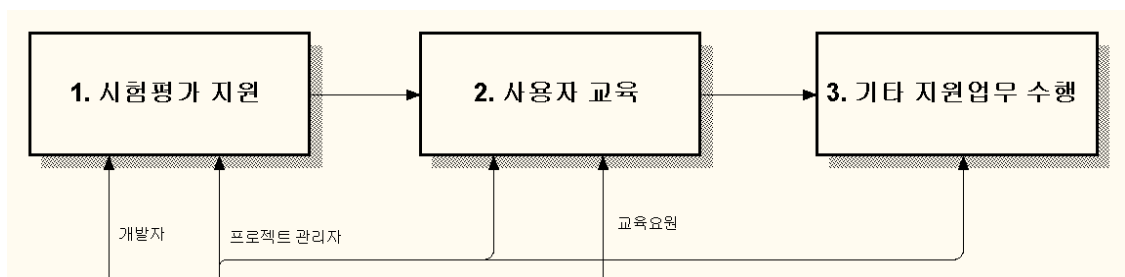
인수 지원 실시(4S21)

개요

사용자 주관의 시험평가를 지원하고 시스템 운용자, 사용자 및 관리자 등을 대상으로 교육을 실시한다. 그 외에 산출물의 납품 및 검수지원 등을 수행한다.

구분	작업 구성 항목
입력물	<ul style="list-style-type: none"> 계약 관련 문서 유스케이스 정의서 시스템 통합 테스트 기술서 사용자 지침서 운용자 지침서
산출물	해당 사항 없음
참여자	<ul style="list-style-type: none"> 개발자 시험평가에 필요한 자료를 제공하며, 발생한 오류를 수정하여 재평가가 가능하도록 지원한다. 교육요원 사용자 지침서, 운용자 지침서 또는 별도의 교육자료 등을 통해 교육을 실시한다. 프로젝트 관리자 시험평가 지원을 총괄하고 계획에 따라 교육이 수행되는지를 확인하며 프로젝트 종결을 위한 업무를 수행한다.
관련기법	해당 사항 없음

절차 흐름도



인수 지원 실시(4S21)

절차 설명

1. 시험평가 지원

사용자 주관으로 실시하는 시험평가가 원활히 수행될 수 있도록 지원한다. 국방 분야에서는 일반적으로 개발시험평가와 운용시험평가가 인수를 위한 시험평가로 실시된다.

- 시험평가 담당자 교육
시험평가를 수행하게 될 사용자를 대상으로 시스템에 대한 소개 및 기능별 사용법 및 시험평가 방법 등을 교육한다.
- 시험평가 관련 자료 제공
사용자측의 요청이 있을 경우 시스템 통합 테스트 기술서를 기반으로 시험평가에 필요한 절차와 테스트 케이스 및 입력 데이터와 예상 결과에 관련된 자료를 제공한다.
- 오류사항 수정
시험평가 결과 요구사항에 부합되지 않거나 오류가 발생한 경우, 해당 사항을 수정하고 재평가할 수 있도록 지원한다.

2. 사용자 교육

제안요청서, 제안서 또는 사업수행계획서 등에서 제시된 교육계획을 토대로 시스템의 사용자, 관리자 및 운용자에 대한 교육을 실시한다.

- 사용자 교육
시스템에 대한 소개 및 기능별 사용법 등 향후 시스템을 통해 업무를 수행하는데 있어 필요한 사항에 대해 교육을 실시한다. 많은 사용자들을 대상으로 교육이 실시되므로 동영상 등을 통한 온라인 교육 또는 집체 교육 등의 방법을 고려한다.
- 관리자 교육
시스템 관리자를 대상으로 시스템 공통 기능 및 지원 기능과 시스템 관리방법 등에 대한 교육을 실시한다.
- 운용자 교육
운용자를 대상으로 시스템의 운용, 정비, 복구 등에 대한 시스템 중심의 교육과 하드웨어, 네트워크 및 소프트웨어 등 상용 제품에 대한 기본 지식 교육을 실시한다. 필요시 상용 제품에 대한 교육은 교재 및 기자재가 준비되어 있는 별도의 전문 교육 기관을 통해 실시할 수 있다.

■ 인수 지원 실시(4S21)

- 유지보수 요원 교육

계약에 관련 내용이 반영되어 있을 경우 사업관리기관이 지정한 유지보수 요원을 대상으로 시스템의 업무 프로세스 개념, 시스템 유지보수 절차, 개발 관련 기본 지식 등에 대한 교육을 실시한다.

3. 기타 지원업무 수행

시험평가 및 사용자 교육이 완료되면 프로젝트의 종결을 위해 산출물의 납품 및 검수지원 업무를 수행한다.

- 산출물 납품 및 검수지원

제안요청서에 명시된 산출물에 대해 현행화를 실시한 후 사용자에게 납품하며 검수를 위한 지원업무를 수행한다.

■ 작업 수행시 고려사항

■ 인수 지원 실시 시 고려사항

- 인수 지원 업무는 시스템 개발 프로세스보다는 사업관리를 위한 프로세스에 해당된다. 따라서 인수 지원 업무의 세부 사항에 대해서는 ‘국방전력발전업무 훈령’ 또는 ‘국방정보체계사업관리지시’ 등의 규정/지침을 참조한다.

인수 지원 실시(4S21)

[서식]

해당 사항 없음

[항목 설명]

해당 사항 없음

국방정보기술아키텍처 활용 가이드

국방정보기술아키텍처 활용 가이드

■ 개 요

국방아키텍처프레임워크(MND-AF)를 기반으로 작성된 국방정보기술아키텍처는 국방 정보화의 청사진으로 국방 분야 시스템의 현행 아키텍처와 목표 아키텍처로 구성되어 있다.

현행 아키텍처와 목표 아키텍처는 시스템이 제공하는 기능, 기반 체계 구성, 다른 시스템과의 인터페이스, 데이터 모델 등에 대해 현 시점에서의 모습과 미래 시점에서 개선될 모습이 아키텍처로 작성되어 있어 시스템 개발시 참고할 수 있으며, 시스템 구축 사업을 수행하는 사업수행기관은 시스템 구축과 함께 해당 시스템의 단위 체계 아키텍처를 작성해야 한다.

국방 CBD 방법론을 적용해 시스템을 개발할 때 국방정보기술아키텍처 구축 결과를 참고하거나 시스템 개발 결과를 기반으로 단위 체계 아키텍처를 구축하는데 도움이 되도록 국방 CBD 방법론과 국방정보기술아키텍처 구축시 적용하는 MND-AF간의 비교를 통해 참고 관계를 제시한다.

※ 본 가이드는 국방 CBD 방법론의 표기법과 의미가 일치하거나 유사한 MND-AF의 엔티티를 제시하여 참조할 수 있도록 한다.

■ 활동별 국방정보기술아키텍처 활용 가이드

■ 요구사항 식별(1R1)

요구사항 식별 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 공통관점(AV)의 산출물(통합사전(AV-2))을 용어집 작성에 참고할 수 있다.
- 운용관점(OV)의 산출물(운용 개념도(OV-1), 조직관계도(OV-4), 운용활동모델(OV-5), 개념데이터모델(OV-7b))을 도메인 및 비즈니스 모델을 정의하는데 참고할 수 있다.
- 체계관점(SV) 산출물(체계정의기술서(SV-1a), 체계인터페이스기술서(SV-1b), 체계통신기술서(SV-2), 체계기능정의서(SV-4), 체계데이터교환목록(SV-6), 체계성능목록(SV-7), 물리데이터모델(SV-11), 체계기반구조기술서(SV-12))을 현행 시스템 분석시 참고할 수 있다.

※ 해당 시스템의 목표 아키텍처를 참조하고 현행 시스템 분석(1R12)만 국방정보기술아키텍처의 현행 아키텍처를 참조한다.

국방정보기술아키텍처 활용 가이드

작업ID	작업명	국방정보기술아키텍처 활용 가이드
1R11	도메인 모델링	<ul style="list-style-type: none"> · 운용개념도(OV-1)의 운용개념도 템플릿을 시스템 도메인 정의 및 다른 시스템간의 관계 식별시 참조 · 통합사전(AV-2)의 용어를 용어집 작성시 참조
1R12	현행 시스템 분석	<ul style="list-style-type: none"> · 체계정의기술서(SV-1a)의 체계 도메인과 체계를 현행 시스템 도메인과 현행 시스템을 정의하는데 참조 · 체계인터페이스기술서(SV-1b)의 체계노드, 체계인터페이스를 현행 타 시스템과의 인터페이스를 정의하는데 참조 · 체계통신기술서(SV-2)의 통신장비, 네트워크를 현행 네트워크 구성 정의시 참조 · 체계기능정의서(SV-4)의 체계기능을 현행 시스템의 주요 기능 정의시 참조 · 체계데이터교환목록(SV-6)의 체계데이터교환(필요시 체계 데이터요소)을 현행 타 시스템과의 인터페이스 내역 정의시 참조 · 체계성능목록(SV-7)의 체계구성요소, 성능요소, 구성 요소 별 성능 측정을 현행 시스템의 성능 정의시 참조 · 물리데이터모델(SV-11)의 물리엔티티타입(필요시 물리엔티티속성, 물리데이터도메인)을 현행 시스템 자료구조 정의시 참조 · 체계기반구조기술서(SV-12)의 하드웨어(하드웨어 인터페이스 포함), 체계분산환경, 체계분산구조, 시스템소프트웨어, 패키지소프트웨어를 현행 시스템의 하드웨어 및 소프트웨어 구성시 참조
1R13	비즈니스 모델링	<ul style="list-style-type: none"> · 조직관계도(OV-4)의 조직, 조직관계를 비즈니스 액터 식별시 참조 · 운용개념도(OV-1)의 업무대기능, 업무중기능 및 운용활동모델(OV-5)의 업무소기능을 비즈니스 패키지 식별시 참조 · 운용활동모델(OV-5)의 단위과제, 운용활동을 비즈니스 프로세스 및 비즈니스 액션 식별시 참조 · 개념데이터모델(OV-7b)의 개념엔티티를 비즈니스 개념 식별시 참조
1R14	요구사항 명세	<ul style="list-style-type: none"> · 직관적인 참조관계는 없으나 해당 시스템의 목표 아키텍처를 요구사항 명세서 활용

국방정보기술아키텍처 활용 가이드

■ 아키텍처 정의(1R2)

아키텍처 정의 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 체계관점(SV)의 산출물(체계정의기술서(SV-1a), 체계통신기술서(SV-2), 체계기반구조기술서(SV-12))을 시스템 아키텍처 정의시 참고할 수 있다.

※ 해당 시스템의 목표 아키텍처를 참조한다.

작업ID	작업명	국방정보기술아키텍처 활용 가이드
1R21	시스템 아키텍처 정의	<ul style="list-style-type: none"> • 체계정의기술서(SV-1a)의 체계 도메인과 체계를 시스템 도메인과 시스템을 정의하는데 참조 • 체계통신기술서(SV-2)의 통신장비, 네트워크를 네트워크 구성 정의시 참조 • 체계기반구조기술서(SV-12)의 하드웨어(하드웨어 인터페이스 포함), 체계분산환경, 체계분산구조, 시스템소프트웨어, 패키지소프트웨어를 시스템의 하드웨어 및 소프트웨어 구성시 참조
1R22	표준 지침 수립	<ul style="list-style-type: none"> • 직관적인 참조관계는 없으나 해당 시스템의 목표 아키텍처를 기술 표준 식별시에 활용

■ 요구사항 분석(1R3)

요구사항 분석 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 체계관점(SV)의 산출물(체계인터페이스기술서(SV-1b), 체계통신기술서(SV-2), 체계기능정의서(SV-4), 체계데이터교환목록(SV-6))을 연동 소요 분석시 참고할 수 있다.

※ 해당 시스템의 목표 아키텍처를 참조한다.

작업ID	작업명	국방정보기술아키텍처 활용 가이드
1R31	유스케이스 모델링	<ul style="list-style-type: none"> • 직관적인 참조관계는 없으나 체계기능정의서(SV-4)의 체계 기능을 유스케이스 식별시 활용

국방정보기술아키텍처 활용 가이드

작업ID	작업명	국방정보기술아키텍처 활용 가이드
1R32	연동 소요 분석	<ul style="list-style-type: none"> · 체계인터페이스기술서(SV-1b)의 체계노드, 체계인터페이스를 연동 대상 시스템 식별 및 주요 연동 항목을 정의하는데 참조 · 체계통신기술서(SV-2)의 통신장비, 네트워크를 연동 기반 환경 분석시 참조 · 체계기능정의서(SV-4)의 체계기능과 외부소스목적지간 관계를 연동 항목 도출시 참조 · 체계데이터교환목록(SV-6)의 체계데이터교환(필요시 체계데이터요소)을 연동 내역 정의시 참조

■ 자료 구축 준비(1R4)

자료 구축 준비 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 체계관점(SV)의 산출물(물리데이터모델(SV-11))을 자료 전환 대상 분석시 참고할 수 있다.
※ 해당 시스템의 현행 아키텍처를 참조한다.

작업ID	작업명	국방정보기술아키텍처 활용 가이드
1R41	자료 구축 계획	<ul style="list-style-type: none"> · 물리데이터모델(SV-11)의 물리엔티티타입(필요시 물리엔티티속성, 물리데이터도메인)을 자료 전환 대상 식별시 참조

■ 테스트 준비(1R5)

테스트 준비 활동은 MND-AF와의 직관적인 참조 관계가 없다.

■ 개략 설계(2D1)

개략 설계 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 운용관점(OV)의 산출물(논리데이터모델(OV-7c))을 데이터 모델링시 참고할 수 있다.
※ 해당 시스템의 목표 아키텍처를 참조한다.

국방정보기술아키텍처 활용 가이드

작업ID	작업명	국방정보기술아키텍처 활용 가이드
2D15	데이터 모델링	<ul style="list-style-type: none"> · 논리데이터모델(OV-7c)의 논리엔티티타입, 논리엔티티속성을 자료 전환 대상 식별시 참조 ※ 논리데이터도메인도 참조 가능하나 데이터 도메인은 MDR을 준용

■ 상세 설계(2D2)

상세 설계 활동 수행에 참고할 수 있는 MND-AF의 산출물을 각 관점 기준으로 정리하면 다음과 같다.

- 체계관점(SV)의 산출물(체계데이터교환목록(SV-6), 체계보안기술서(SV-13))을 연동 설계 및 시스템 보안 설계시 참고할 수 있다.
※ 해당 시스템의 목표 아키텍처를 참조한다.

작업ID	작업명	국방정보기술아키텍처 활용 가이드
2D24	연동 설계	· 체계데이터교환목록(SV-6)의 체계데이터교환, 체계데이터요소를 타 시스템과의 인터페이스 내역 정의시 참조
2D26	시스템 보안 설계	· 체계보안기술서(SV-13)의 보안영역, 보안요소, 하부보안요소, 보안정책을 기반으로 시스템의 각 구성요소를 맵핑

■ 이후 활동

이후 2D3 시스템 설치 준비, 3T1 구현, 3T2 단위 테스트, 3T3 소프트웨어 통합 및 테스트, 3T4 시스템 통합 및 테스트, 3T5 지침서 작성, 4S1 시스템 설치, 4S2 인수 지원 활동은 MND-AF와의 직관적인 참조 관계가 없다.

■ 감 수 관	정보화기획관	고위공무원	김 재 민
	정보화정책과장	서 기 관	유 동 주
■ 검토/편집	정보화제도 담당	사 무 관	정 영 임
■ 과제 연구	한국국방연구원 국방정보체계관리단	연구위원	김 주 필
		선임전문연구원	임 종 수
		선임전문연구원	윤 병 권
		선임전문연구원	이 동 준
		선임전문연구원	민 지 영

국방 CBD 방법론
(v2.0 사용자 지침서 1권)

2010년 5월 18일 인 쇄
2010년 5월 18일 발 행
편집 : 국방부 정보화기획관실

서 면 의 결 서

- '10-4차 국방CIO 실무협의회 상정안건을 아래와 같이 의결함.
- 기 간 : '10. 4. 21. ~ 4. 28.
- 심의안건 : 국방 CBD 방법론 개정안(v2.0)
- 심의결과 : “가결”

구 분	직 책	직계급	성 명	심의결과		비 고 (의 견)
				가	부	
위원장	정보화기획관	고위 공무원	김재민	○		
위 원	합참 지통부 연동평가과장	대령	양종휴	○		-CBD산출물관리를 위한 관련 규정 발전 -전문관리조직 양성 및 임무부여 -산출물 평가를 위한 후속절차 및 제도정립
	육군 정보화기획실 응용체계과장	대령	이종훈	○		-활동을 현업을 고려하여 개선시켰고 소규모 테일러링 가이드 및 유지보수 산출물 선정기준 등을 발전시켰음
	해군 정보화기획실 정보화기획과장	대령	이종현	○		-
	공군 정보화기획실 정보화발전과장	중령	김성표	○		- 개선안을 개발 및 유지 보수에 선 적용중이며 개선사항 발생 시 상신 예정임
	국방전산정보원 자원정보화과장	서기관	박준빈	○		-
	방위사업청 획득기반과장	대령	최광묵	○		-