

대분류/20
정보통신

중분류/01
정보기술

소분류/02
정보기술개발

세분류/02
응용SW엔지니어링

능력단위/01

NCS학습모듈

요구사항 확인

LM2001020201_16v3



교육부

학습 1 현행 시스템 분석하기

학습 2 요구사항 확인하기

학습 3 분석모델 확인하기

1-1. 현행 시스템 파악

학습 목표

- 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다.

필요 지식 /

① 현행 시스템 파악의 개요

1. 현행 시스템 파악의 정의 및 목적

(1) 현행 시스템 파악의 정의

현행 시스템이 어떤 하위 시스템으로 구성되어 있는지, 제공하는 기능이 무엇인지, 다른 시스템들과 어떤 정보를 주고받는지, 어떤 기술요소를 사용하고 있는지, 사용하고 있는 소프트웨어 및 하드웨어는 무엇인지, 네트워크는 어떻게 구성되어 있는지 등을 파악하는 활동이다.

(2) 현행 시스템 파악의 목적

이를 통하여 향후 개발하고자 하는 시스템의 개발범위 및 이행방향성 설정에 도움을 주는 것이 목적이다.

2. 현행 시스템 파악 절차

아래와 같이 3단계로 구분하여 수행해야 할 활동들에 대하여 기술한다.

(1) 1단계

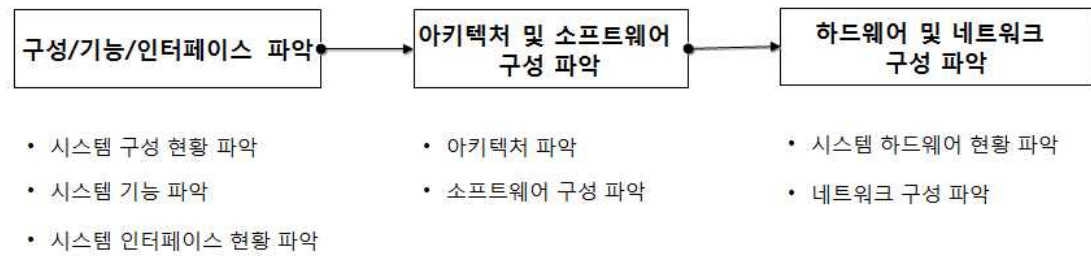
현행 시스템의 구성, 기능, 인터페이스 현황을 파악하는 단계

(2) 2단계

현행 시스템의 아키텍처 및 소프트웨어 구성 현황을 파악하는 단계

(3) 3단계

현행 시스템의 하드웨어 및 네트워크 구성 현황을 파악하는 단계



[그림 1-1] 현행 시스템 파악 절차

② 현행 시스템 구성/기능 및 인터페이스

1. 현행 시스템 구성 현황

(1) 현행 시스템 구성 현황의 정의

현행 시스템 구성 현황은 조직의 주요 업무를 처리하는 기간 업무와 이를 지원하는 지원 업무로 구분하여 기술한 것이다.

(2) 현행 시스템 구성 현황 작성 시 고려 사항

각 업무에 속하는 단위 업무 정보시스템들의 명칭, 주요 기능들을 명시함으로써 조직 내 존재하는 모든 정보시스템의 현황을 파악하도록 한다.

2. 기능 현황

(1) 기능 현황의 정의

단위 업무 시스템이 현재 제공하고 있는 기능을 기술한 것이다.

(2) 기능 현황 작성 시 고려 사항

단위 업무 시스템에서 제공하는 기능들을 주요 기능과 하부 기능으로 구분하여 계층 형으로 표시한다.

3. 인터페이스 현황

(1) 인터페이스 현황의 정의

단위 업무 시스템이 다른 단위 업무 시스템과 주고받는 데이터의 종류, 데이터 형식, 프로토콜, 연계유형, 주기 등을 명시한 것이다.

(2) 인터페이스 현황 작성 시 고려 사항

중요한 고려 사항으로는 어떤 형식(format)으로 데이터를 주고받는지(XML, 고정 포맷, 가변 포맷 등), 어떤 통신규약(TCP/IP, X.25 등)을 사용하고 있고, 연계유형(EAI, FEP 등)은 무엇인지 등이 있다.

③ 현행 시스템 아키텍처 및 소프트웨어

1. 현행 시스템 아키텍처 구성도

(1) 현행 시스템 아키텍처 구성도의 정의

기간 업무를 수행하기 위하여 계층별로 어떠한 기술 요소들을 사용하고 있는지 최상위 수준에서 그림으로 표현한 것이다.

(2) 현행 시스템 아키텍처 구성도 작성 시 고려 사항

단위 업무 시스템별로 아키텍처가 다른 경우에는 가장 핵심이 되는 기간 업무 처리 시스템을 기준으로 한다.

2. 소프트웨어 구성도

(1) 소프트웨어 구성도의 정의

단위 업무 시스템의 업무 처리를 위해 설치되어 있는 소프트웨어들의 제품명, 용도, 라이선스 적용 방식, 라이선스 수를 명시한 것이다.

<표 1-1> 시스템별 소프트웨어 현황 및 작성 예시

구분	시스템 명	SW 제품명	용도	라이선스 적용 방식	라이선스 수
기간 업무	단위 업무 A 시스템	Apache Tomcat	WAS	오픈 소스 Apache License	
		MySQL	데이터베이스	GPL 또는 상용	

(2) 소프트웨어 구성도 작성 시 고려 사항

시스템 구축 시 인프라 구축 비용에서 하드웨어 비용뿐만 아니라 소프트웨어 비용이 적지 않기 때문에, 상용 소프트웨어의 경우에는 라이선스 적용 방식의 기준(사이트, 서버, 프로세서, 코어(core), 사용자 수 등)과 보유한 라이선스 수량 파악이 중요하다.

④ 하드웨어 및 네트워크

1. 하드웨어 구성도

(1) 하드웨어 구성도의 정의

단위 업무 시스템들이 어디에 위치하고 있는 서버에서 운용되고 있는지 서버의 주요 사양(CPU 처리 속도, 메모리 크기, 하드디스크의 용량 등)과 수량, 이중화가 적용되어 있는지 여부를 명시한 것이다.

<표 1-2> 시스템별 하드웨어 현황 및 작성 예시

구분	시스템명	서버 용도	제품명	주요 사양	수량	이중화
기간 업무	단위 업무 A 시스템	AP	서버	CPU 4core:	1	N
		서버	제품명 A1	Memory: 8GB HDD: 500GB		
		DB	서버	CPU 8core:	1	N
		서버	제품명 A2	Memory: 16GB HDD: 1TB		

• 하드디스크 드라이브(HDD: Hard Disk Drive), 테라바이트(TB: Tera Byte), 기가바이트(GB: Giga Byte)

(2) 하드웨어 구성도 작성 시 고려 사항

이중화는 기간 업무의 서비스 기간, 장애 대응 정책에 따라 필요성 여부가 결정되며, 현행 시스템에서 이중화가 적용된 경우에는 목표 시스템에서도 이중화가 필요한 경우가 대부분이며, 이에 따라 인프라 구축 기술 난이도 및 비용 증가 가능성이 존재한다.

2. 네트워크 구성도

(1) 네트워크 구성도의 정의

업무 처리 시스템들이 어떠한 네트워크 구성을 가지고 있는지 그림으로 표현한 것이다.

<표 1-3> 네트워크 현황 및 작성 예시

위치	용도	장비 제품명	주요 사양	수량	비고
전산 센터	방화벽	제품명		2	
IDC	라우터	제품명		1	

인터넷 데이터 센터(IDC: Internet Data Center)

(2) 네트워크 구성도 작성 시 고려 사항

네트워크 구성도의 작성을 통해 서버의 위치, 서버 간의 네트워크 연결 방식을 파악할 수 있다. 네트워크 구성도는 조직 내 서버들의 물리적인 위치 관계 파악, 조직 내 보안 취약성 분석 및 대응, 네트워크 장애 발생 추적 및 대응 등의 다양한 용도로 활용될 수 있다.

수행 내용 / 현행 시스템 분석서 작성 및 검토하기

재료·자료

- 사용자 매뉴얼, 운영자 매뉴얼, 현행 담당자 면담 기록

기기(장비·공구)

- 컴퓨터, 프로젝터, 문서 작성 프로그램

안전·유의사항

- 분석 및 발표는 정해진 시간 안에 종료될 수 있도록 한다.

수행 순서

① 현행 시스템 관련 자료를 수집한다.

1. 현행 시스템 관련 자료를 수집하기 위하여
 - (1) 정보시스템 구성/기능 및 인터페이스
 - (2) 현행 시스템 아키텍처 및 소프트웨어
 - (3) 하드웨어 및 네트워크와 같이 3개의 팀으로 나눈다.
2. 각 팀별로 수집할 자료의 목록을 정하고, 자료를 수집하기 위하여 현행 시스템 담당자가 제시한 자료와 면담 기록에 필요자료의 존재여부를 파악한다.

<표 1-4> 수집할 자료 목록 예시

구분	취득할 자료	설명
정보시스템 구성/기능 및 인터페이스	정보시스템 구성, 기능 구성도, 인터페이스 현황	해당 자료가 없을 경우에는, 운영자 매뉴얼, 사용자 매뉴얼, 현행 시스템 구축 제안서/분석서/설계서에서 관련 자료를 발췌할 수 있다.
현행 시스템 아키텍처 및 소프트웨어	현행 시스템 아키텍처 구성도, 소프트웨어 구성도	관련 자료가 문서로 존재하지 않는 경우, 현행 시스템 담당자와 면담 등을 통해서 작성해야 한다.
하드웨어 및 네트워크	하드웨어 구성도, 네트워크 구성도	자료를 취득할 경우, 자료의 출처, 수령한 시점, 현행 시스템과 동기화 여부, 담당자의 의견 등을 기록한다.

② 수집한 자료를 분석한다.

1. 수집된 자료에 산발적으로 존재하는 정보들을 취합/정제 작업을 실시한다. 즉, 중복되거나 유효하지 않은 오래된 정보, 현행 시스템과 일치하지 않는 정보들은 삭제되어야 한다.
2. 관련된 자료가 존재하지 않거나, 현행 시스템 담당자가 일부 내용만 알고 있는 경우에는 분석할 수 있는 내용까지 정리한다. 불명확한 부분은 해당 내용을 명시하여 분석 및 설계 단계에서 참조하여 구체적으로 조사할 수 있게 한다.
3. 각 팀은 자신이 담당하고 있는 분야와 관련된 현행 시스템의 이슈, 문제점을 파악한다.

<표 1-5> 분석할 내용 예시

구분	작성할 산출물	분석할 내용
정보시스템 구성/기능 및 인터페이스	정보시스템 구성/ 기능 구성도	정보시스템의 구분, 정보시스템 명, 주요 기능, 정보시스템의 기능 구성
	인터페이스 현황	송신 시스템, 수신 시스템, 연계 방식, 주기, 연 동 데이터 형식, 주요 연동 데이터
현행 시스템 아키텍처 및 소프트웨어	현행 시스템 아키텍 처 구성도	운용체제의 종류, 적용 프레임워크, 계층별 적 용 기술, 데이터 연동 방식, 데이터 저장소, 외 부 시스템 연계 방식
	소프트웨어 구성도	단위 시스템별 필요 소프트웨어의 제품명, 용도, 라이선스 적용 방식, 라이선스 수
하드웨어 및 네트워크	하드웨어 구성도	단위 시스템별 서버 제품명, 용도, CPU, 메모리, 하드디스크 용량, 수량, 이중화 적용 여부
	네트워크 구성도	네트워크 장비의 용도, 제품명, 수량, 주요 사 양, 위치, 연동 프로토콜

③ 분석한 결과를 산출물로 작성한다.

1. 분석한 결과를 활용 서식을 참조하여 산출물을 작성한다. 각 산출물의 예는 <표 1-6>과 같다.

<표 1-6> 정보시스템 구성 현황 및 작성 예시

구분	시스템 명	시스템 내용	비고
기간 업무	단위 업무 A 시스템	업무 A를 처리하기 위하여 기능 A1, 기능 A2 등을 제공하는 시스템	
	단위 업무 B 시스템	업무 B를 처리하기 위하여 기능 B1, 기능 B2 등을 제공하는 시스템	
지원 업무	지원 업무 C 시스템	지원 업무 C를 처리하기 위하여 기능 C1, 기 능 C2 등을 제공하는 시스템	

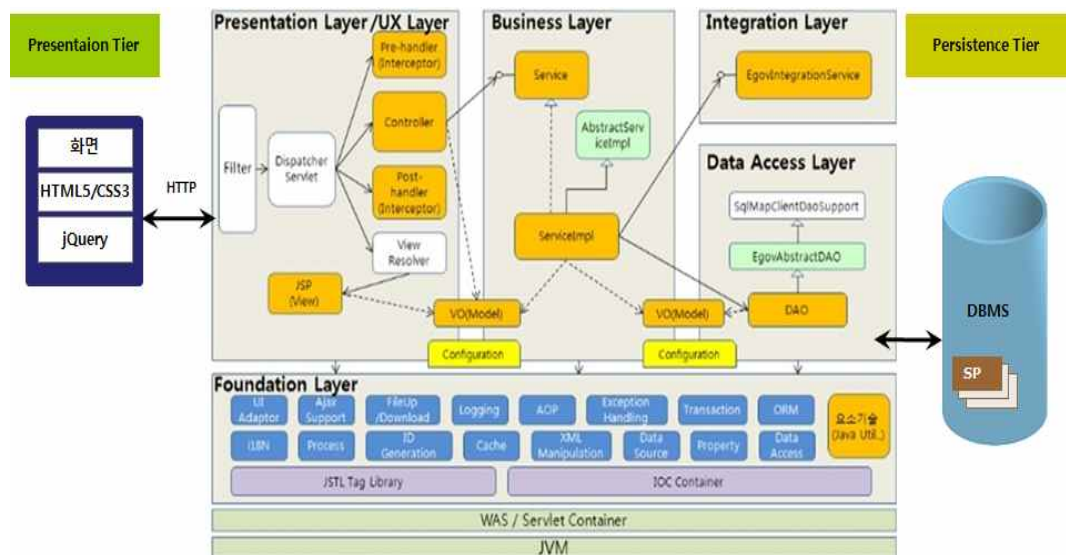
<표 1-7> 정보시스템 기능 구성도 예시

송신 시스템	수신 시스템	연동 데이터	연동 형식	통신 규약	연계 유형	주기
단위 업무 A 시스템	대외 기관 A 시스템	연체 정보	XML	TCP/IP	EAI	일 배치
단위 업무 A 시스템	대외 기관 B 시스템	부도 정보	XML	X.25	FEP	수시

<표 1-8> 시스템 인터페이스 현황 및 작성 예시

시스템 명	기능 L1	기능 L2	기능 L3	비고
단위 업무 A 시스템	기능 1	하부 기능 11	세부 기능 111	
			세부 기능 112	
		하부 기능 12	세부 기능 121	
			세부 기능 122	
	기능 2	하부 기능 21	세부 기능 211	

- 전사 애플리케이션 통합(EAI: Enterprise Application Integration), 조직 내 응용 프로그램 통합 솔루션
- 프론트 엔드 프로세서(FEP: Front End Processor), 대외계 서버



전자정부 프레임워크 v3.0

출처: <http://www.egovframe.go.kr/wiki/doku.php?id=egovframework:mcom> 모바일 공통 컴포넌트란

[그림 1-2] 전자정부 모바일 공통 컴포넌트 적용 아키텍처 예시

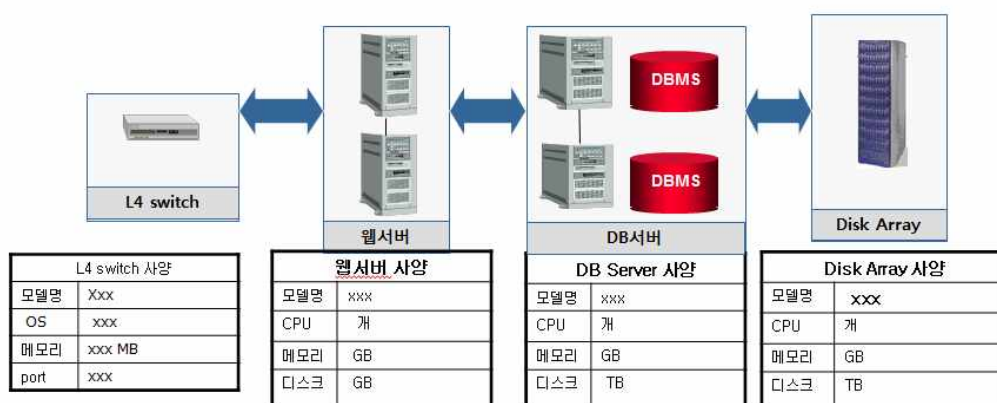
※ [그림 1-2]에 대한 설명

전자정부 모바일 공통 컴포넌트는 효율적인 전자정부 기반 시스템의 구축 운영을 통해 전자정부의 서비스 품질 및 정보화 투자 효율 향상의 기반을 확보하고, 모바일 전자정부 서비스에 대한 접근성을 제고하기 위해 모바일 표준 프레임워크 기반으로 개발된 공통 컴포넌트로서 주요 구성 요소는 다음과 같다.

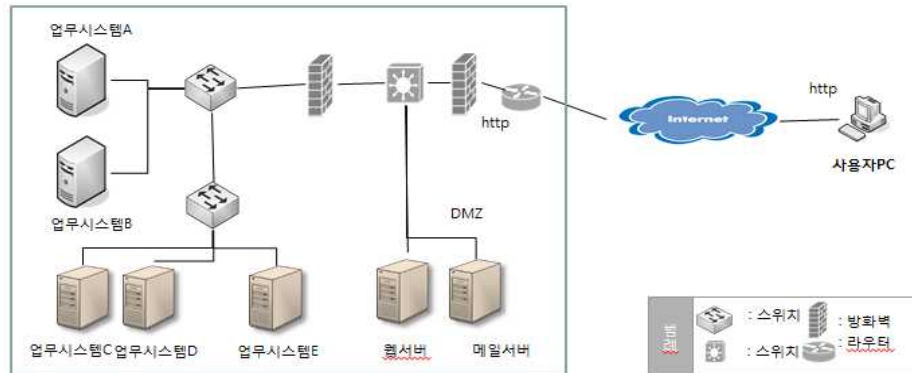
- 공통 기반 계층(Foundation Layer): 실행 환경 서비스 간에 공통적으로 사용되는 기능을 제공한다.
 - 화면 처리 계층(Presentation Layer/UX Layer): 업무 처리 서비스와 사용자 간의 인터페이스를 담당하는 서비스로 사용자 화면 구성 및 사용자 입력 정보 검증 등의 기능을 지원한다.
 - 업무 처리 계층(Business Layer): 업무 프로그램의 업무 로직을 담당하는 서비스로 업무 흐름 제어, 트랜잭션 관리, 에러 처리 등의 기능을 제공한다.
 - 데이터 처리 계층(Data Access Layer): 업무 프로그램에서 사용할 수 있도록 데이터에 대한 CRUD 기능을 지원한다.
 - 연계 통합 계층(Integration Layer): 타 시스템과의 연동 기능을 지원한다.
- 기타 자세한 내용은 [그림 1-2] 출처 인터넷 주소(URL)를 방문하면 찾아볼 수 있다.



[그림 1-3] 소프트웨어 구성도 예시 (MS로부터 이용허락을 받았습니다.)



[그림 1-4] 하드웨어 구성도 예시 (MS로부터 이용허락을 받았습니다.)



[그림 1-5] 네트워크 구성도 예시 (MS로부터 이용허락을 받았습니다.)

2. 산출물에는 현행 시스템의 이슈, 문제점을 포함한다.

4. 작성된 산출물에 대하여 검토를 수행한다.

1. 각 팀별로 작성된 산출물들은 서로 밀접하게 관련되어 있으므로, 팀별로 작성된 산출물을 발표하면 다른 팀은 자신의 산출물과 관련하여 의견을 제시한다.
2. 다른 팀의 검토 의견을 반영하여 산출물을 수정한다.

수행 tip

- 현행 시스템의 모든 내용을 상세히 분석하게 되면 많은 시간과 노력이 소요되므로, 향후 구축될 시스템의 목적에 맞추어 분석 범위와 수준을 정하는 것이 필요하다.

1-2. 개발 기술 환경 정의

학습 목표

- 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다.
- 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.

필요 지식 /

① 개발 기술 환경

개발 기술 환경을 정의할 때 고려할 사항을 [그림 1-6]과 같이 운영체제, DBMS, 미들웨어, 오픈소스 순으로 살펴본다.

본 학습에서는 모든 미들웨어에 대하여 기술하지는 않으며 자주 사용되는 웹 애플리케이션 서버(WAS: Web Application Server)를 선정할 경우 고려해야 할 사항에 대해서 기술하고, 오픈소스 사용 시 주의해야 할 내용과 저작권 관련 정보를 제시한다.



[그림 1-6] 개발 기술 환경 정의 시 고려 사항

1. 운영체제 주요 특징 및 고려 사항

(1) 운영체제의 정의

운영체제(OS: Operating System)는 하드웨어와 소프트웨어 리소스를 관리하고 컴퓨터 프로그램을 위한 공통 서비스를 제공하는 소프트웨어를 의미한다. 관련 사이트 참고 (https://en.wikipedia.org/wiki/Operating_system)

(2) 운영체제의 종류 및 특징

주요 운영체제로는 마이크로소프트 윈도우(Microsoft Windows), 유닉스(UNIX), 리눅스(Linux), 아이오에스(iOS), 안드로이드(Android) 등이 있다.

<표 1-9> 운영체제 종류 및 특징

종류	저작자	predecessor	비용 및 라이선스	주요 용도
Windows	Microsoft	OS/2, MS-DOS	유상, 다양한 라이선스 정책	중소 규모 서버, 개인용 PC, Tablet PC, Embedded System
UNIX	IBM(AIX), HP(HP-UX), SUN(Solaris)	UNIX System V, SunOS	유상, 다양한 라이선스 정책	대용량 처리, 안정성이 요구되는 서버, Server, NAS, Workstation
Linux	Linus Torvalds	Linux kernel	무료, GNU GPLv2	중대 규모 서버
iOS	Apple	OS X NeXTSTEP, BSD	하드웨어에 번들 (Buddle)	스마트폰, 태블릿 PC, Music 플레이어 등
Android	Google	Linux	무료, Apache 2.0, GNU GPLv2	스마트폰, 태블릿 PC

관련 사이트 참고(https://en.wikipedia.org/wiki/Comparison_of_operating_systems)

자바 가상 머신(JVM: Java Virtual Machine)은 다양한 하드웨어 및 운영체제에서 자바(Java) 언어로 작성된 애플리케이션을 수행하기 위한 사양(JVM Specification)의 구현체(Implementation)를 의미한다. 오라클(Oracle)이 자바(Java) 상표를 소유하고 있으며, 핫스팟(Hotspot) 구현체와 클래스 라이브러리(Class Library) 구현체를 배포하고 있다. 아이비엠(IBM)의 J9, 오라클(Oracle)의 JRockit(이전 BEA System 제공) 등 벤더별로 여러 자바 가상 머신(JVM: Java Virtual Machine) 구현체를 배포하고 있다. 관련 사이트 참고(https://en.wikipedia.org/wiki/Java_virtual_machine)

(3) 고려 사항

정보시스템 구축 시 운영체제 관련 요구사항을 식별할 때 고려해야 할 사항은 다음과 같다.

<표 1-10> 운영체제 관련 요구사항 식별을 위해 고려해야 할 사항

구분	내용
신뢰도	장기간 시스템을 운영할 때 운영체제 고유의 장애 발생 가능성 특정 응용프로그램의 메모리 누수로 인한 성능 저하 및 재기동 운영체제의 보안상 허점으로 인한 반복적인 패치 설치를 위한 재기동 운영체제의 버그 등으로 인한 패치 설치를 위한 재기동
성능	대규모 동시 사용자 요청 처리 대규모 및 대량 파일 작업 처리 지원 가능한 메모리 크기(32bit, 64bit)
기술 지원	공급 벤더들의 안정적인 기술 지원 다수의 사용자들 간의 정보 공유 오픈 소스 여부(Linux)
주변 기기	설치 가능한 하드웨어 다수의 주변 기기 지원 여부
구축 비용	지원 가능한 하드웨어 비용 설치할 응용프로그램의 라이선스 정책 및 비용 유지 및 관리 비용 총 소유 비용(TCO)

- (가) 일반적으로 리눅스(Linux) 기반 시스템이 하드웨어 및 소프트웨어 소유 비용이 가장 적게 소요된다.
- (나) 유지 및 관리 비용 측면에서는 윈도우(Windows) 기반 시스템이 강점을 가진다.
- (다) 안정적이고 신뢰할 수 있으며 대용량 처리를 위해서는 유닉스(UNIX) 기반 시스템이 선호되고 있다.
- (라) 32bit 운영체제는 4GB 메모리까지 액세스 가능(사용자 메모리는 2GB)하지만, 64bit 운영체제에서는 4GB 이상의 메모리에 액세스 가능하며 구체적인 한계는 운영체제의 종류 및 버전에 따라 다양하다.
- (마) 시스크(CISC: Complex Instruction Set Computer) 설계 방식이 적용된 인텔의 x86 아키텍처 기반 칩을 사용하고 있는 하드웨어는 윈도우(Windows)나 리눅스(Linux)를 운영체제로 설치할 수 있으며, 리스크(RISC: Reduced Instruction Set Computer) 설계 방식이 적용된 칩들은 유닉스(UNIX) 운영체제를 설치한다.
- (바) 에이치피(HP)와 인텔(Intel)이 협력해서 만든 아이테니엄 아키텍처(IA: Itanium Architecture)-64 칩은 여러 운영체제를 지원한다.
- (사) 리스크(RISC) 설계 방식이 적용된 암(ARM) 칩은 스마트폰이나 태블릿에 주로 채용되고 있으며, 아이오에스(iOS), 안드로이드(Android) 등의 운영체제를 지원하고 있다. 관련 사이트 참고

(https://en.wikipedia.org/wiki/Complex_instruction_set_computing)

2. DBMS 주요 특징 및 고려 사항

(1) DBMS의 정의

사용자, 다른 애플리케이션, 데이터베이스와 상호 작용하여 데이터를 저장하고 분석하기 위한 컴퓨터 소프트웨어 애플리케이션으로, 데이터베이스 생성, 조회, 변경 등의 관리가 주요 기능이다. 관련 사이트 참고(<https://en.wikipedia.org/wiki/Database>)

(2) DBMS의 종류 및 특징

<표 1-11> DBMS 종류 및 특징

종류	저작자	비용 및 라이선스	주요 용도
Oracle	Oracle	상용	대규모, 대량 데이터의 안정적인 처리
IBM DB2	IBM	상용	대규모, 대량 데이터의 안정적인 처리
Microsoft SQL Server	Microsoft	상용	중소 규모 데이터의 안정적인 처리
MySQL	MySQL AB, Oracle	GPL 또는 상용	오픈 소스 RDBMS
SQLite	D. Richard Hipp	Public Domain 저작권 보호 만료	스마트폰, 태블릿 PC 등의 임베디드 데이터베이스(Embedded Database) 용
MongoDB	MongoDB Inc.	GNU AGPL v3.0	오픈 소스 NoSQL 데이터베이스
Redis	Salvatore Sanfilippo	BSD	오픈 소스, 메모리, 키-값(Key-Value) 데이터베이스

(3) 고려 사항

정보시스템 구축 시 DBMS 관련 요구사항 식별을 위하여 고려할 사항은 <표 1-12>와 같다.

<표 1-12> DBMS에 관하여 고려할 사항

구분	내용
가용성	장기간 시스템을 운영할 때 장애 발생 가능성 DBMS의 버그 등으로 인한 패치 설치를 위한 재기동 백업 및 복구 편의성 DBMS 이중화 및 복제 지원
성능	대규모 데이터 처리 성능(분할 테이블의 지원 여부) 대량 거래 처리 성능 다양한 튜닝 옵션 지원 비용 기반 최적화 지원 및 설정의 최소화
기술 지원	공급 벤더들의 안정적인 기술 지원 다수의 사용자들 간의 정보 공유 오픈 소스 여부
상호 호환성	설치 가능한 운영체제 종류 다양한 운영체제에서 지원되는 JDBC, ODBC
구축 비용	라이선스 정책 및 비용 유지 및 관리 비용 총 소유 비용(TCO)

관련 사이트 참고

(<http://www.dbguide.net/knowledge.db?cmd=view&boardUid=126000&boardConfigUid=19>)

인 메모리 DB(In-memory DB)는 속도가 빠르지만 물리적인 메모리 용량 확장에 한계가 있다.

3. 미들웨어의 주요 특징 및 고려 사항

(1) 미들웨어의 정의

운영체제와 소프트웨어 애플리케이션 사이에 위치하는 미들웨어(Middleware)는 소프트웨어 애플리케이션에게 운영체제가 제공하는 서비스를 추가 및 확장하여 제공하는 컴퓨터 소프트웨어를 말한다. (<https://en.wikipedia.org/wiki/Middleware>) 여기에서는 미들웨어 중 웹 애플리케이션 서버(WAS: Web Application Server)에 대해서 알아본다.

(2) 웹 애플리케이션 서버(WAS: Web Application Server)의 정의

동적인 웹 사이트, 웹 애플리케이션, 웹 서비스의 개발을 지원하기 위하여 설계된 소프트웨어로서 데이터 접근, 세션 관리, 트랜잭션 관리 등을 위한 라이브러리를 제공하고 있다. 관련 사이트 참고(https://en.wikipedia.org/wiki/Web_application_framework)

(3) 웹 애플리케이션 서버(WAS: Web Application Server)의 종류 및 특징

<표 1-13> 웹 애플리케이션 서버(WAS: Web Application Server) 종류 및 특징

종류	벤더	비용 및 라이선스	주요 용도
GlassFish	GlassFish Community	CDDL, GPL	NetBeans 개발 툴과 연동하여 사용
JBoss	Red Hat	LGPL	JBoss 기반 오픈 소스 제품들을 이용하는 경우
Jetty	Eclipse Foundation	Apache 2.0, EPL	빠른 처리 속도가 요구되는 경우
JEUS	TmaxSoft	상용	대량의 안정적인 거래 처리가 요구되며 적시의 기술 지원이 필요한 경우
Resin	Caucho Technology	GPL, 상용	빠른 처리 속도가 요구되는 경우
WebLogic	Oracle Cor.	상용	대량의 안정적인 거래 처리가 요구되는 경우
WebSphere	IBM	상용	경우

관련 사이트 참고(https://en.wikipedia.org/wiki/Comparison_of_application_servers)

(4) 고려 사항

정보시스템 구축 시 웹 애플리케이션 서버(WAS: Web Application Server) 관련 요구사항 식별을 위하여 고려할 사항은 다음과 같다.

<표 1-14> 웹 애플리케이션 서버(WAS: Web Application Server)에 관하여 고려할 사항

구분	내용
가용성	장기간 시스템을 운영할 때 장애 발생 가능성 안정적인 트랜잭션 처리 WAS의 버그 등으로 인한 패치 설치를 위한 재기동 WAS 이중화 지원
성능	대규모 거래 요청 처리 성능 다양한 설정 옵션 지원 가비지 컬렉션(GC: Garbage Collection)의 다양한 옵션
기술 지원	공급 벤더들의 안정적인 기술 지원 다수의 사용자들 간의 정보 공유 오픈 소스 여부
구축 비용	라이선스 정책 및 비용 유지 및 관리 비용 총 소유 비용(TCO: Total Cost of Ownership)

4. 오픈 소스 사용에 따른 고려 사항

(1) 오픈 소스의 정의

오픈 소스(Open Source)는 소스 코드를 공개해 누구나 특별한 제한 없이 그 코드를 보고 사용할 수 있는 오픈 소스 라이선스를 만족하는 소프트웨어를 말한다. 관련 사이트 참고

(https://ko.wikipedia.org/wiki/%EC%98%A4%ED%94%88_%EC%86%8C%EC%8A%A4_%EC%86%8C%ED%94%84%ED%8A%B8%EC%9B%A8%EC%96%B4)

(2) 오픈 소스 사용 시 고려 사항

오픈 소스를 사용하는 경우에는 라이선스의 종류, 사용자 수, 기술의 지속 가능성 등을 고려해야 한다. 라이선스의 종류 등 자세한 내용은 한국저작권위원회의 OLIS 사이트(<https://www.olis.or.kr>)를 참조한다. 어떠한 오픈 소스를 사용해야 라이선스의 문제가 없을지 판단이 어려운 경우에는 전자정부 표준 프레임워크에서 사용 중인 오픈 소스 소프트웨어를 참조할 수 있다. 관련 사이트 참고

(<http://www.egovframe.go.kr/EgovOSS.jsp?menu=1&submenu=2&leftsub=3>)

② 시스템 용량산정 방법

시스템을 구축하기 위하여 필요한 하드웨어 규모는 부록에 제시한 한국정보통신기술협회의 “정보시스템 하드웨어 규모산정 지침” TTA-KO-10.0292 관련 내용을 참조하여, CPU 용량, 메모리 용량, 디스크 용량을 산정할 수 있다.

수행 내용 / 개발 기술 환경 요구사항 파악하기

재료·자료

- 현행 시스템 담당자 면담 기록, 시스템 용량산정 엑셀 파일

기기(장비·공구)

- 컴퓨터, 프로젝터, 문서 작성 프로그램, 엑셀 프로그램

안전·유의사항

- 분석 및 발표는 정해진 시간 안에 종료될 수 있도록 한다.

수행 순서

① 기술 환경 정의를 위한 자료를 수집한다.

1. 수집할 자료의 목록을 정하고, 자료를 수집하기 위하여 현행 시스템 담당자가 제시한 자료와 면담 기록에 필요 자료의 존재 여부를 파악한다.

<표 1-15> 온라인 트랜잭션 처리(OLTP: OnLine Transaction Processing)를 위한 기초 자료 조사 항목

항목	설명
시스템 구축 형태	단독 시스템(Single System), 고가용성 시스템(HA System), 병렬 구성 여부
사용자 수	전체 사용자 수, 동시 사용자 비율, 동시 사용자당 평균 질의 수(1일), 가동 시간 중 피크타임의 시간, 연간 사용자 증가율
트랜잭션 수	연간 트랜잭션 수, 1일 평균 트랜잭션 수, 피크타임 트랜잭션 수, 예상 연간 트랜잭션 증가율
온라인 업무	검색, 갱신, 삽입, 삭제 별 레코드 크기 및 전체 건수
배치 업무	온라인 업무에 대한 배치 업무 비중, 배치 업무 구분, 대량 배치 기준으로 데이터 건수 및 길이
데이터베이스	데이터 크기(초기, 1년차, 2년차, 3년차 및 이후 데이터 증가율), 데이터 중 이미지, 사운드, 텍스트 파일의 비율, 인덱스 테이블의 초기 크기 및 3년 내 크기, 가장 큰 테이블의 레코드 건수, 데이터베이스 크기
데이터 백업	데이터 백업, 데이터 백업 서버의 운영 여부, 백업 장치의 접속 패턴, 백업 데이터의 양
운영 시간	운영 시간 7x24 여부

2. 시스템 사용 현황 파악을 위하여 <표 1-15>와 같은 자료를 조사한다.

<표 1-16> WEB/WAS를 위한 기초 자료 조사 항목

항목	설명
시스템 용도 및 서비스 형태	웹 페이지만 제공, 트랜잭션이 빈번하지 않은 웹 서비스(데이터베이스 연계), 트랜잭션이 빈번한 웹 서비스(데이터베이스 연계)
시스템의 구성 형태	1계층, 2계층, 3계층
접속자 수	평균 접속자 수(24시간 기준), 최고 접속자 수(1시간), 연간 접속자 증가율
사용률	동시 사용자 수, 사용자당 오퍼레이션 수, 이미지 파일과 사운드 파일의 크기, 웹 페이지 크기, 허용 응답 시간
업무 중요도 및 긴급도	중요도(상중하), 긴급도(상중하)
백 엔드 상호 작용 형태	읽기 전용(Read only), 업데이트(Update), 온라인 트랜잭션 처리(OLTP)
SSL 사용 여부	안전한 통신이 필요한지 여부

* 위의 기초 자료 조사 항목 중에서 현업 담당자 면담 기록에 존재하는 부분만 발췌하여 시스템 용량산정에 활용한다. 존재하지 않는 항목에는 기본 값을 적용한다.

② 조사 자료 분석 및 개발 기술 환경을 결정한다.

1. 조사한 자료를 이용하여 운영체제, DBMS, 웹 애플리케이션 서버(WAS: Web Application Server) 등을 결정한다.

<표 1-17> 개발 기술 환경 결정을 위한 고려 사항

항목	고려 사항
운영체제	시스템 구축 예산이 적은 경우(예 2천만 원 이하)에는 유닉스(UNIX)를 도입하기 어렵다. 리눅스(Linux)는 비용이 저렴하나 유지 관리를 위한 기술 인력을 보유하거나 별도의 계약을 체결해야 한다. 유닉스(UNIX)는 안정적이고 대량의 처리가 가능하고 기술 지원이 용이하나 비용이 많이 소요된다. 윈도즈(Windows)는 유지 관리 기술 인력 확보가 용이하고 유닉스(UNIX)에 비해 상대적으로 비용이 저렴하나, 대부분의 대용량 처리 서버에 설치할 수 없다.
DBMS	상용 DBMS의 경우 안정적이며 확장성이 뛰어나고 기술 지원을 받기 용이하나 비용이 많이 소요된다. 오픈 소스 DBMS의 경우 비용이 저렴하나, 관련된 기술력을 자체적으로 확보할 필요가 있다. 일반적으로 많이 사용되고 있는 DBMS(상용 또는 오픈 소스 모두)를 선택하면 관련 기술 인력 및 기술 자료를 확보하기 용이하고 문제 해결이 용이하다.
웹 애플리케이션 서버 (WAS: Web Application Server)	표준 규격을 준수하는 웹 애플리케이션 서버(WAS)의 경우 개발용과 운영용을 구분하여 사용할 수 있다. 개발용은 가볍고 빠른 오픈 소스 웹 애플리케이션 서버(WAS)를 선택할 수 있다. 상용 웹 애플리케이션 서버(WAS)의 경우에는 안정적이며, 대량 처리가 검증되어 있고 기술지원을 받기가 용이하다. 오픈 소스 웹 애플리케이션 서버(WAS)의 경우 일반적으로 널리 사용하는 웹 애플리케이션 서버(WAS)를 선택하는 것이 바람직하다.

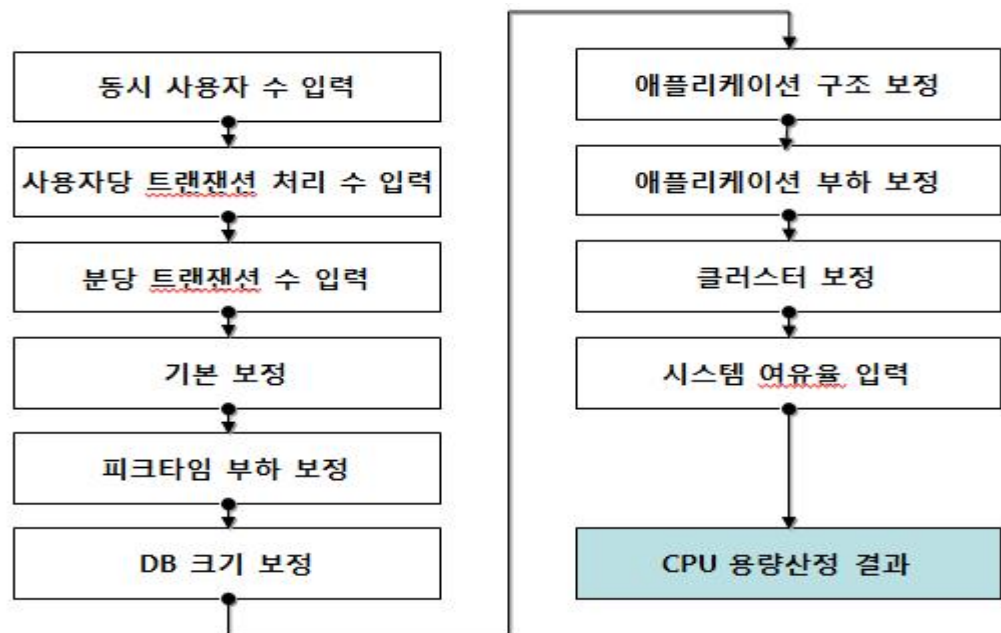
2. 조사한 자료를 이용하여 시스템 용량을 산정한다.

수집 및 분석된 온라인 트랜잭션 처리(OLTP: Online Transaction Processing), WEB/WAS 기초 자료 조사 항목의 값을 시스템 용량산정 엑셀 파일에 입력하여 CPU, 메모리, 디스크 용량을 계산한다.

(1) CPU 용량산정

(가) OLTP/Batch/데이터베이스 서버

온라인 트랜잭션 처리(OLTP: Online Transaction Processing)/일괄작업(Batch)/데이터베이스 서버의 CPU 용량을 산정하기 위한 상세 내용은 본 학습모듈의 부록을 참조한다. 구체적인 CPU 용량 산정 절차는 [그림 1-7]과 같다.



[그림 1-7] CPU 용량산정 - OLTP/Batch/DB 서버 산정 절차

위의 절차대로 시스템 용량을 산정할 수 있도록 한국정보화진흥원에서는 엑셀(Excel) 파일을 제공하고 있다.

(http://www.nia.or.kr/bbs/board_view.asp?BoardID=201111281321074458&id=1536&Order=&search_target=&keyword=&Flag=) 아래는 이 파일의 온라인 트랜잭션 처리(OLTP) 서버의 CPU 용량산정 시트(Sheet)이다. WEB/WAS 용량산정, 메모리 용량산정, 디스크 용량산정 시트도 제공하고 있다.

OLTP 서버의 CPU 용량산정

순서	구분	계산값	입력값	비고
1	동시 사용자수	1000	1000	
2	트랜잭션 처리수	5000	5개	
3	기본 TPMC 보정	6500	1.3	
4	Peak Time 보정	7800	1.2	
5	데이터베이스 크기 보정	1	DB Size: 1 Max Row: 0.3	
6	어플리케이션 복잡도 보정	0.9	형태: 단순검색 테이블 수: 40	
7	사용자 복잡성 보정	1	동시수: 30 접속수: 100	
8	어플리케이션 구조 보정	1.1	구조1: 3-Tier 구조2: DB Only 응답시간: 3 값: 1.1	
9	어플리케이션 부하 보정	1.7	Batch 크기: Light Data 유형: 동일한 데이터	
10	네트워크 보정	1.1		
11	클러스터 보정	1.3	클러스터 유무: Y 1.3	
12	여유율 보정	1.2	1.2	
TPMC		22,527		

한국정보화진흥원 제공 정보시스템 용량산정 도구(엑셀 파일)

[그림 1-8] OLTP 서버의 CPU 용량산정 엑셀 파일

(나) WEB/WAS 서버

WEB/WAS 서버의 CPU 용량을 산정하기 위한 상세 내용은 본 학습모듈의 부록을 참조한다. 구체적인 CPU 용량 산정 절차는 [그림 1-9]와 같다.



[그림 1-9] CPU 용량산정 - WEB/WAS 서버 산정 절차

(2) 메모리 용량산정

서버의 메모리를 산정하기 위한 상세 내용은 본 학습모듈의 부록을 참조한다. 구체적인 메모리 용량산정 절차는 [그림 1-10]과 같다.



[그림 1-10] 메모리 용량산정 절차

(3) 디스크 용량산정

시스템의 디스크 용량을 산정하기 위한 상세 내용은 본 학습모듈의 부록을 참조한다. 디스크 용량산정 절차는 [그림 1-11]과 같다.



[그림 1-11] 디스크 용량산정 절차

③ 요구사항 정의서, 목표 시스템 구성도에 반영한 후 검토한다.

1. 운영체제, DBMS, 웹 애플리케이션 서버(WAS: Web Application Server), 시스템 용량 산정 결과를 요구사항 정의서, 목표 소프트웨어 구성도, 목표 하드웨어 구성도에 결과를 반영하여 수정한다.
2. 팀별로 작성된 산출물 최종 결과를 발표하면, 다른 팀은 의견을 제시한다.
3. 다른 팀의 검토 의견을 반영하여 산출물을 수정한다.

수행 tip

- 개발 기술 환경을 정의하기 위한 운영체제, DBMS, WAS, 오픈 소스 라이선스 등의 모든 내용을 다루기에는 그 내용이 방대하므로 상세 내용은 참고 문헌이나 인터넷 주소(URL)를 제시하는 것이 바람직하다.

학습 1	현행 시스템 분석하기
학습 2	요구사항 확인하기
학습 3	분석모델 확인하기

2-1. 요구사항 정의

학습 목표

- 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용 소프트웨어의 요구사항을 확인할 수 있다.
- 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다.

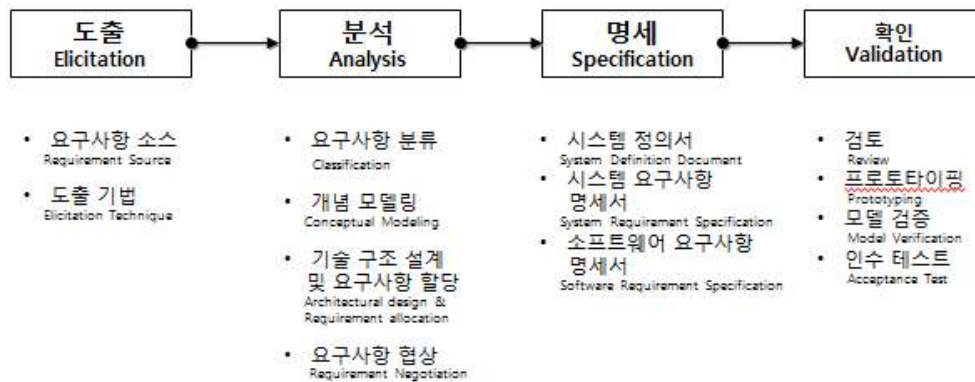
필요 지식 /

① 요구공학 개요

요구공학(Requirements Engineering)이란 요구사항을 정의하고, 문서화하고, 관리하는 프로세스를 의미한다. (https://en.wikipedia.org/wiki/Requirements_engineering)

1. 요구사항 개발 프로세스

소프트웨어공학 지식체계(SWEBOK: SoftWare Engineering Body of Knowledge)에서는 이 프로세스를 요구사항 도출(Elicitation), 분석(Analysis), 명세(Specification), 확인(Validation)으로 구분하고 있다. (<http://www.computer.org/web/swebok>)



[그림 2-1] 요구사항 개발 프로세스

(1) 요구사항 도출(Requirement Elicitation)

- (가) 요구사항 도출은 소프트웨어가 해결해야 할 문제를 이해하는 첫 번째 단계로서 요구사항이 어디에 있고, 어떻게 수집할 것인가와 관련되어 있다.
- (나) 이 단계에서 이해관계자(Stakeholder)가 식별되고, 개발 팀과 고객 사이의 관계가 만들어진다.
- (다) 이 단계에서는 다양한 이해관계자와 효율적인 의사소통이 중요하다.

(2) 요구사항 분석(Requirement Analysis)

- (가) 요구사항들 간 상충되는 것을 해결하고, 소프트웨어의 범위를 파악하며, 소프트웨어가 환경과 어떻게 상호 작용하는지 이해한다.
- (나) 시스템 요구사항을 정제하여 소프트웨어 요구사항을 도출한다.

(3) 요구사항 명세(Requirement Specification)

- (가) 요구사항 명세란 체계적으로 검토, 평가, 승인될 수 있는 문서를 작성하는 것을 의미한다.
- (나) 시스템 정의, 시스템 요구사항, 소프트웨어 요구사항을 작성한다.

(4) 요구사항 확인(Requirement Validation)

- (가) 분석가가 요구사항을 이해했는지 확인(Validation)이 필요하고, 요구사항 문서가 회사의 표준에 적합하고 이해 가능하며, 일관성이 있고, 완전한지 검증(Verification)하는 것이 중요하다.
- (나) 이해관계자들이 문서를 검토해야 하고, 요구사항 정의 문서들에 대해 형상 관리를 해야 하는데, 일반적으로 요구사항 관리 툴을 이용한다.
- (다) 리소스가 요구사항에 할당되기 전에 문제를 파악하기 위하여 검증을 수행한다.

위와 같은 요구사항 개발 프로세스 중에서 요구사항 확인하기와 관련된 단계는 분석 및 검증 단계이므로, 필요 지식에서는 도출 및 명세 단계를 생략한 분석 및 검증 단계에 대해서만 기술하기로 한다.

② 요구사항 분석 기법

요구사항 분석을 통해 요구사항을 기술할 때에는 아래의 작업들이 가능하도록 충분하고 정확하게 기술하여야 한다.

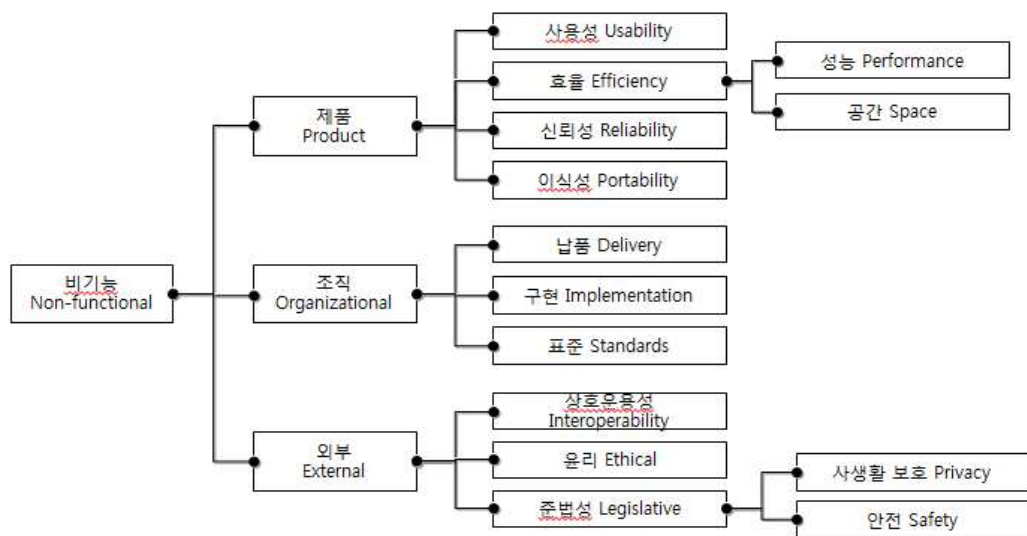
- 요구사항의 확인(Validation)
- 요구사항 구현의 검증(Verification)
- 비용 추정

분석기법으로 요구사항 분류((Requirement Classification), 개념 모델링(Conceptual Modeling), 요구사항 할당((Requirement Allocation), 요구사항 협상((Requirement Negotiation), 정형 분석(Formal Analysis) 등이 있다.

1. 요구사항 분류(Requirement Classification)

요구사항을 다음과 같은 기준으로 분류한다.

- 요구사항이 기능인지 비기능인지
- 요구사항이 하나 이상의 고수준 요구사항으로부터 유도된 것인지 또는 이해관계자나 다른 원천(Source)으로부터 직접 발생한 것인지
- 요구사항이 제품에 관한 것인지 프로세스에 관한 것인지
- 우선순위가 더 높은 것인지 여부
- 요구사항의 범위(요구사항이 소프트웨어에 미치는 영향의 범위)
- 요구사항이 소프트웨어 생명 주기 동안에 변경이 발생하는지 여부



[그림 2-2] 비기능 요구사항의 분류 예시

2. 개념 모델링(Conceptual Modeling)

(1) 개념 모델의 역할

(가) 실세계 문제에 대한 모델링이 소프트웨어 요구사항 분석의 핵심이며, 모델은 문제가 발생하는 상황에 대한 이해를 증진시키고 해결책을 설명한다.

(나) 따라서 개념 모델은 문제 도메인의 엔터티(entity)들과 그들의 관계 및 종속성을 반영한다.

(2) 개념 모델의 종류와 표기법

(가) 유스케이스 다이어그램(Use Case Diagram), 데이터 흐름 모델(Data Flow Model), 상태 모델(State Model), 목표기반 모델(Goal-Based Model), 사용자 인터랙션(User Interactions), 객체 모델(Object Model), 데이터 모델(Data Model) 등과 같은 다양한 모델을 작성할 수 있다.

(나) 대부분의 모델링 표기법은 UML(Unified Modeling Language)을 사용한다.

(3) UML 다이어그램의 사용

(가) 사용 시나리오를 나타내기 위하여 유스케이스 다이어그램이 많이 사용되고 있다.

(나) 구조 다이어그램(Structure Diagram)은 시스템의 정적 구조(Static Structure)와 다양한 추상화 및 구현 수준에서 시스템의 구성 요소, 구성 요소들 간의 관계를 보여 준다.

(다) 행위 다이어그램(Behavior Diagram)은 시스템 내의 객체들의 동적인 행위(Dynamic Behavior)를 보여 주며, 시간의 변화에 따른 시스템의 연속된 변경을 설명해 준다.

3. 요구사항 할당(Requirement Allocation)

(1) 요구사항을 만족시키기 위한 아키텍처 구성 요소를 식별하는 것을 요구사항 할당이라 한다.

(2) 다른 구성 요소와 어떻게 상호 작용하는지 분석을 통하여 추가적인 요구사항을 발견할 수 있다.

4. 요구사항 협상(Requirement Negotiation)

(1) 두 명의 이해관계자가 서로 상충되는 내용을 요구하거나, 요구사항과 리소스, 기능과 비기능 요구사항들이 서로 상충되는 경우, 어느 한 쪽을 지지하기보다는 적절한 트레이드 오프 지점에서 합의가 중요하다.

(2) 요구사항에 우선순위를 부여하는 것은 중요한 요구사항을 필터링할 수 있으며, 요구사항들 간 상충되는 문제를 해결하는 데 사용될 수 있다.

5. 정형 분석(Formal Analysis)

(1) 형식적으로 정의된 시맨틱(Semantics)을 지닌 언어로 요구사항을 표현한다.

(2) 정확하고 명확하게 표현하여 오해를 최소화시킬 수 있다.

(3) 정형 분석(Formal Analysis)은 요구사항 분석의 마지막 단계에서 이루어진다.

③ 요구사항 확인

분석가가 요구사항을 이해했는지 확인(Validation)하는 것이 필요하고, 요구사항 문서가 회사의 표준에 적합하고 이해 가능하고, 일관성이 있고, 완전한지 검증(Verification)하는 것이 중요하다. 이해관계자들이 문서를 검토해야 하고, 요구사항 정의 문서들에 대해 형상 관리를 해야 하는데 일반적으로 요구사항 관리 툴을 이용한다. 리소스가 요구사항에 할당 되기 전에 문제를 파악하기 위하여 검증을 수행한다.

1. 요구사항 확인 기법

(1) 요구사항 검토(Requirement Reviews)

(가) 요구사항 검증의 가장 일반적인 방법으로, 여러 검토자들이 에러, 잘못된 가정, 불명확성, 표준과의 차이 등을 찾아내는 작업을 수행하며, 검토자 그룹을 어떻게 구성하느냐가 중요하다.

(나) 예를 들어, 고객 중심 프로젝트에서는 검토자 그룹에 고객 대표자가 1명 이상 포함되어야 한다.

(다) 검토는 시스템 정의서(System Definition Document), 시스템 사양서(System Specification), 소프트웨어 요구사항 명세서(SRS: Software Requirements Specification Document)를 완성한 시점 등에서 이루어진다.

<표 2-1> IEEE Std 830-1998의 사용자 클래스 기준으로 조직화된 SRS 템플리트

3. 상세 요구사항(Specific Requirements)
3.1 외부 인터페이스 요구사항(External Interface Requirements)
3.1.1 사용자 인터페이스(User Interface)
3.1.2 하드웨어 인터페이스(Hardware Interface)
3.1.3 소프트웨어 인터페이스(Software Interface)
3.1.4 통신 인터페이스(Communication Interface)
3.2 기능 요구사항(Functional Requirements)
3.2.1 사용자 클래스 (User Class) 1
3.2.1.1 기능 요구사항 1.1
...
3.2.m 사용자 클래스(User Class) m
3.2.1.1 기능 요구사항 m.1
...
3.2.1.n 기능 요구사항 m.n
3.3 성능 요구사항(Performance Requirements)
3.4 설계 제약사항(Design Constraints)
3.5 소프트웨어 시스템 속성(Software System Attributes)
3.6 기타 요구사항(Other Requirements)

(2) 프로토타이핑(Prototyping)

- (가) 프로토타이핑은 새로운 요구사항을 도출하기 위한 수단으로, 또한 소프트웨어 요구사항에 대해 소프트웨어 엔지니어가 해석한 것을 확인하기 위한 수단으로 많이 사용된다.
- (나) 프로토타이핑의 장점은 분석가의 가정을 파악하고 잘못된 경우 유용한 피드백을 제공한다는 점, 사용자 인터페이스(User Interface)의 동적인 행위가 문서나 그래픽 모델보다 프로토타입으로 이해하기 쉬운 점, 요구사항의 가변성이 프로토타이핑 이후에 급격히 감소하는 점이다.
- (다) 단점은 사용자의 관심이 핵심 기능에서 멀어지고 프로토타입의 디자인이나 품질 문제로 집중될 수 있으며, 프로토타입 수행 비용이 발생한다는 것이다.
- (라) 잘못된 요구사항을 만족시키기 위하여 자원을 낭비하는 것을 방지할 수 있다는 점에서 프로토타이핑을 긍정적으로 검토할 수 있다.

(3) 모델 검증(Model Verification)

- (가) 분석단계에서 개발된 모델의 품질을 검증할 필요가 있다.
- (나) 예를 들어, 객체 모델의 경우 객체들 사이의 존재하는 의사소통 경로(Communication Path)를 검증(Verify)하기 위하여 정적 분석(Static Analysis)을 수행하는 것이 유용하다.

(4) 인수 테스트(Acceptance Tests)

- (가) 요구사항의 중요한 속성은 최종 제품이 요구사항을 만족시키는지 확인이 가능해야 한다는 것이다
- (나) 각각의 요구사항을 어떻게 확인할 것인지에 대한 계획을 세워야 한다.

요구사항 검증 단계에서 사용되는 기법 이외에 요구사항 품질 검증을 위한 국내 표준을 활용하여 요구사항을 검증할 수 있다. 정보통신단체표준 TTAK.KO-11.0103 “소프트웨어 요구사항 품질 평가 항목”에서는 요구사항 명세의 품질을 객관적이고 정량적으로 평가하기 위한 기준으로 평가 항목을 제시하고 있다. 구체적인 품질 모델, 품질 특성, 품질 특성 평가 항목은 부록을 참조한다.

수행 내용 / 요구사항 검증하기

재료 · 자료

- 현행 시스템 분석서, 사용자 면담 기록, 요구사항 목록, 유스케이스 명세서(요구사항 정의서)

기기(장비 · 공구)

- 컴퓨터, 문서 작성 프로그램, UML 저작 프로그램

안전 · 유의사항

- 실습 시 지도 교사의 지시를 따른다.

수행 순서

- 1 요구사항 목록에 업무 기능에 대한 요구사항이 모두 반영되었는지 확인한다.

〈표 2-2〉 요구사항 목록 예시

ID	이름	유형	품질 속성	우선 순위	중요도	출처	내용
Req-001	회원 ID 유일성	기능	N/A	상	10	김관리	회원은 유일한 ID를 가져야 한다.
Req-002	비밀번호 찾기	기능	N/A	중	5	김관리	회원이 비밀번호를 분실한 경우 비밀번호를 찾기 위한 2가지 이상의 기능을 제공해야 한다.

개별 요구사항에 대하여 〈표 2-3〉과 같은 속성에 대해서 검토한다.

<표 2-3> 요구사항 목록의 속성

요구사항 속성	검토 내용
ID	명명 규칙에 따라 유일성이 확보되는 식별자가 부여되었는지 확인한다.
이름	요구사항 내용을 요약하고, 중복되지 않았는지 확인한다.
유형	기능, 비기능, 제약사항, 기타로 구분되어 있는지 확인한다.
품질 속성	유형이 비기능일 때 품질 속성으로 성능, 가용성, 유지보수성, 신뢰성, 보안성, 유지보수 용이성, 사용 용이성 등이 명시되어 있는지 확인한다.
우선순위	요구사항의 우선순위가 얼마나 높은지, 필수, 선택, 희망 사항 등으로 구분되어 있는지 확인한다.
중요도	요구사항의 중요도가 작성 규칙에 따라 적절한 점수가 부여되어 있는지 확인한다.
출처	요구사항을 낸 이해관계자의 이름이나 관련 문서명이 기술되어 있는지 확인한다.
관련 부서	요구사항과 관련된 조직의 부서명이 기술되어 있는지 확인한다.
전제 조건	요구사항과 관련된 전제 조건이 적절한지 확인한다.
내용	요구사항의 내용이 명확하고 이해하기 쉽게 기술되어 있는지 확인한다.
관련 요구사항	관련된 요구사항이 적절한지 확인한다.
버전	요구사항의 변경 상태에 따라 버전이 관리되고 있는지 확인한다.
수용 여부	검토 예정, 수락, 거부 등 수용 여부 진행 상태가 기술되어 있는지 확인한다.

② 요구사항 정의서의 작성 여부를 확인한다.

1. 요구사항 목록 중 수용 여부가 수락인 경우, 요구사항 정의서인 유스케이스 명세서가 작성되었는지 확인한다. 유스케이스 명세서가 시스템이 어떻게 동작하여야 하는지를 명확하고 구체적으로 기술하고 있는지 검토한다.

<표 2-4> 유스케이스 명세서 예시

항목	내 용
유스케이스 명	로그인한다.
관련 액터	사용자, 회원, 비회원, 관리자
개요	사용자는 공인 인증서를 이용하고 비회원은 메일 주소 및 비밀번호를 이용하여 로그인한다.
사전 조건	회원은 사전에 회원 가입이 되어 있어야 한다.
사후 조건	회원은 읽고 쓰기 권한, 비회원은 읽기 권한만 부여되어야 한다.
정상 흐름	1. 로그인 화면에서 회원은 공인 인증서 로그인을 선택한다. 1-1. 공인 인증서에 유효한 비밀번호를 입력한다. 1-2. 회원 등급에 맞는 권한을 부여한다. 2. 로그인 화면에서 비회원은 비회원 로그인을 선택한다. 2-1. 비회원은 메일 주소 및 비밀번호를 제공하여 로그인한다.

항목	내 용
	2-2. 비회원 등급에 맞는 권한을 부여한다.
대안 흐름	1. 공인 인증서 로그인을 선택한 후 유효한 비밀번호가 제시되었으나, 사전에 회원 가입이 되어 있지 않은 경우, 회원 가입이 필요하다는 메시지를 보여 준다. 1-1. 회원 가입 페이지로 화면을 전환한다.
예외 흐름	1. 회원이 입력한 공인 인증서의 비밀번호가 유효하지 않은 경우, 로그인 실패 메시지와 실패 횟수를 보여 준다. 2. 로그인 실패 횟수가 연속하여 5회 이상이 되면 해당 회원의 로그인 잠금을 설정하여 1시간 이내에는 로그인이 불가능하게 한다. 3. 회원이 관리자에게 로그인 잠금 해제를 요청하면, 관리자는 유효한 회원인지 검증을 한 후 잠금을 해제한다.
비기능적 요구사항	사용자가 가장 많은 피크 타임 시에도 3초 이내 로그인이 완료되어야 한다.

③ 비기능적 요구사항이 명확하게 도출되었는지 확인한다.

1. 기능적 동작으로 표현되지 않는 시스템 특성, 품질, 제약사항 등 비기능적 요구사항이 명확하게 도출되었는지 검토한다.
2. 성능, 가용성, 사용 용이성, 유지보수 용이성, 안전성, 보안성 등에 대한 요구사항이 문서화되었는지 확인한다.
3. 비기능적 요구사항의 품질 목표가 정량적으로 정의되어 있으며 검증 가능한 것인지 검토한다.
4. 정상 업무 수행, 시스템 부하가 높은 경우 등 상황에 따라 응용시스템의 성능(응답 시간, 데이터 처리량 등) 목표치가 구체적으로 정해져 있는지 검토한다.

④ 타 시스템과 연계 및 인터페이스에 대한 요구사항이 적절히 도출되었는지 확인한다.

1. 타 시스템 또는 하위 시스템 등과의 모든 인터페이스 요구사항이 정의되어 있는지 확인한다.
2. 연계할 기관과 연계에 대한 합의나 계약이 체결되어 있는지 확인한다.
3. 인터페이스 구분(내부/외부)이 명확하게 정의되어 있는지, 인터페이스 주기, 방법, 제공자, 요청자 등이 명확하게 정의되어 있는지 확인한다.
4. 인터페이스 기능 구현과 관련된 제약사항 및 조건들이 기술되어 있는지 검토한다.

수행 tip

- 요구사항 목록의 속성은 필요에 따라 변경 및 추가 될 수 있다. 요구사항에 대한 처리 프로세스나 평가 속성을 추가하는 경우도 있다.

2-2. 요구사항의 시스템화 타당성 분석

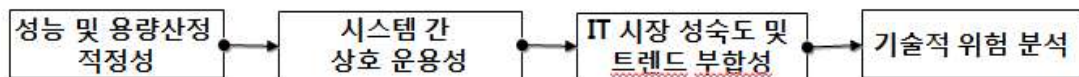
학습 목표

- 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.

필요 지식 /

① 요구사항의 기술적 타당성 검토

“정보화사업 사전타당성분석 방법론 연구”(한국전산원 2004)에서 “기술적 타당성 분석에서는 적용기술의 적합성 및 기술실현의 가능성이 분석의 핵심적 내용이 된다.”라고 명시하면서, 성능 및 용량산정의 적정성, 시스템 간 상호 운용성, IT 시장 성숙도 및 트렌드 부합성, 기술적 위험 분석을 언급하고 있다.



[그림 2-3] 요구사항 기술적 타당성 검토 4단계

1. 성능 및 용량산정의 적정성

“1-2. 개발 기술 환경 정의 ② 시스템 용량산정 방법”에서 목표 시스템의 용량이 산정되면, 과거 유사 프로젝트 경험치를 적용하여 필요시 재조정 한 후, 성능 관련 비기능 요구사항과 비교하여 적정성 여부를 판단한다.

2. 시스템 간 상호 운용성

상호 운용성이란 다른 목적을 지닌 2개 이상 시스템들이 상호 간 정보 및 서비스를 교환하면서 효과적으로 운용될 수 있는 시스템의 능력을 의미한다(한국전산원 2004). 요구사항 중에서 목표 시스템이 조직 내외 타 시스템과의 연동을 요구하는 경우, 상호 운용이 가능한지 여부를 판단해야 한다.

3. IT 시장 성숙도 및 트렌드 부합성

시스템 구축 시 요구되는 영역별 정보 기술들의 시장 성숙도 및 발전 방향을 파악하고, 요구사항이 이에 부합하는지 판단해야 한다. 시장 성숙도가 낮거나, 발전 방향에 부합되지 않는 기술들은 향후 더 이상 사용되지 않을 가능성이 높아 시스템의 유지보수가 어려운 상황이 발생할 수 있다.

4. 기술적 위험 분석

요구사항을 만족시키기 위하여 적용한 기술의 복잡성, 검증 여부, 의존성 등에 대하여 위험 발생 가능성, 영향도를 파악한다.

<표 2-5> 기술적 위험 분석

기술의 특성	내 용
복잡성	기술의 안정성, 시장성, 개방성을 저해하는 모든 요소 하드웨어, 소프트웨어, 솔루션의 적용이 아키텍처와 불일치
검증 여부	적용 기술에 대한 조직 내 무경험 외부 지원 불가능
의존성	특허 및 라이선스에 따른 문제 특정 업체 기술에 대한 의존

수행 내용 / 요구사항의 시스템화 타당성 분석하기

재료·자료

- 현행 시스템 분석서, 사용자 면담 기록, 요구사항 목록, 유스케이스 명세서(요구사항 정의서), 시스템 용량산정 결과 엑셀(Excel) 파일

기기(장비·공구)

- 컴퓨터, 문서 작성 프로그램, 엑셀(Excel) 프로그램

안전·유의사항

- 실습 시 지도 교사의 지시를 따른다.

수행 순서

① 요구사항 목록에 타당성 분석을 위한 속성을 추가하고 타당성 분석 결과를 기록한다.

1. 요구사항 목록에 <표 2-6>과 같이 타당성 분석을 위한 속성(성능/용량, 시스템 간 상호 운용성, 시장 성숙도 및 트렌드 부합성, 기술 복잡성, 기술 검증 여부, 기술 의존성)을 추가한다.

<표 2-6> 요구사항의 시스템화 타당성 분석을 위한 서식

ID	이름	성능/ 용량	시스템 간 상호 운용성	시장 성숙도 및 트렌드 부합성	기술 복잡성	기술 검증 여부	기술 의존성
Req-001	요구사항명

일반적으로 엑셀(Excel)을 이용하여 분석 결과를 기록할 때 위와 같은 형태를 사용하나, 문서 편집기 툴에서 타당성 분석 결과 예시를 기록하기 어렵기 때문에 <표2-7>과 같이 형태를 변경하여 요구사항의 시스템화 타당성 분석 결과를 제시한다.

<표 2-7> 요구사항의 시스템화 타당성 분석 결과 예시

항목	내용
요구사항	모든 거래의 응답 시간은 1초 이내에 처리되어야 한다.
성능/용량	<ul style="list-style-type: none"> • 용량산정 결과 도입될 장비의 tpmC 계산을 위하여, 동시 사용자 수를 100,000, 사용자당 트랜잭션 수를 3으로 하여 계산한 결과 대략 5,310,000tpmC가 산정되었으며, 이에 따른 장비 도입 계획이 만들어져 있으나, 도입된 장비들을 어떻게 구성하느냐에 따라 전체 처리 성능이 달라질 수 있다. • 따라서 도입된 장비들의 운영을 위한 실제 구성이 완료된 이후에 대표적인 거래들을 이용하여 성능 테스트가 필요하다. • 하지만, 성능 테스트를 위한 대표적인 거래들을 선정할 때 데이터의 누적에 따라 처리 시간이 증가하는 거래의 경우에는 응답 시간을 1초 이내로 보장하기 어렵다. 따라서 응답 시간 1초 이내는 단순 조회성 거래로 제한하여야 하며, 장전문 거래의 경우 페이지 처리를 추가해야 한다.
시스템 간 상호운용성	<ul style="list-style-type: none"> • 타 시스템과 연계가 이루어지는 거래의 경우에는 타 시스템의 성능을 고려해야 한다. 따라서 거래 응답 시간을 1초 이내로 보장하기 어렵다. • 내부 시스템에서 타 시스템으로 응답을 요청하기까지 시간, 타 시스템의 처리 시간, 타 시스템으로부터 응답을 받아서 사용자까지 처리하는 시간으로 구분해서 목표치를 설정해야 보다 정확한 성능 목표가 설정될 수 있다.
시장 성숙도 및 트렌드 부합성	대상 아님
기술 검증 여부	<ul style="list-style-type: none"> • 이번 시스템 구축 시 도입될 실시간 거래 모니터링 제품은 거래별 모든 정보를 파일에 기록하고 있기 때문에 거래의 처리 속도 저하가 예상되며, 이에 따른 거래 처리 속도 저하에 대한 구체적인 수치를 제공하고 있지 않기 때문에 해당 제품에 따른 성능 저하가 예상되며, 이는 거래 시간 1초 이내 목표치에 심각한 영향을 줄 수 있다.
기술 의존성	<ul style="list-style-type: none"> • 실시간 거래 모니터링 제품의 문제 발생 시 해당 제품을 제공하는 벤더 사에 기술적인 의존성이 발생할 수 있다.

- (1) 요구사항 유형이 비기능이고, 품질 속성이 성능인 경우, 요구되는 성능과 용량이 시스템 용량산정 결과 엑셀(Excel) 파일과 비교하여 적합한지 검토한다.
- (2) 타 시스템과 연계 및 인터페이스에 대한 요구사항인 경우, 현행 시스템 분석서상의 시스템 인터페이스 현황과 네트워크 구성도, 사용자 면담 기록 등을 파악하여 상호 운용이 가능한지 검토한다.
- (3) 요구사항을 만족시키기 위하여 요구되는 기술이 시장 성숙도가 낮고 트렌드와 일치하지 않아 관련된 소프트웨어나 하드웨어 환경 구축이 어렵고 관련 전문 지식이나 전문가를 조달하기 어려운지 검토한다.
- (4) 요구사항을 만족시키기 위하여 적용한 기술의 복잡성 때문에 시스템을 구축하기 위한 인력 자원 및 기간이 계획을 초과하여 추가적인 비용이 발생하는지 여부를 검토한다.
- (5) 요구사항을 만족시키기 위하여 적용한 기술을 기반으로 개발이나 문제 해결 등을 조직 내 인적 자원으로 수행하기 어려워 외부 지원을 받아야 함으로써 추가적인 비용이 발생하고 시스템 성공 가능성이 저하되는지 여부를 검토한다.
- (6) 요구사항을 만족시키기 위하여 적용한 기술을 사용하기 위해서 특허 사용료를 지불해야 하거나, 추가적인 라이선스를 구매하거나, 특정 업체의 상용 기술 지원에 의존해야 하는지 여부를 검토한다.

② 요구사항의 시스템화 타당성 분석 결과를 요구사항 관련 이해관계자가 검증한다.

1. 요구사항의 시스템화 타당성 분석 결과를 요구사항 관련 이해관계자에게 배포하여 사전 검토를 요청한다.
 시스템화 타당성 분석 결과 검증을 위한 회의 이전에 관련 이해관계자가 사전 검토를 하도록 함으로써 검증 작업이 효과적 효율적으로 진행될 수 있도록 만들기 위한 것이다.
2. 관련 이해관계자가 모여 시스템화 타당성 분석 결과를 검증한다.
 관련 이해관계자 간 이견이 있을 수 있으므로 모든 관련 이해관계자가 검증 회의에 참여하는 것이 중요하다.
3. 타당성 분석 결과에 이견이 있는 경우 프로젝트 관리자(Project Manager)의 중재 하에 합의를 도출한다.
 요구사항의 시스템화 타당성 분석 결과가 관련 이해관계자 간 이견이 있는 상태로 남아 있지 않도록 프로젝트 관리자의 중재 하에 타당성 분석 결과의 이견을 해소하도록 한다.

③ 이해관계자 검증을 거친 타당성 분석 결과를 확인하고 배포 및 공유한다.

1. 이해관계자 검증을 거친 타당성 분석 결과를 의사 결정자가 확인한다.

(1) 의사 결정자의 확인을 통해 요구사항의 시스템화 타당성 분석 결과에 대한 검증
작업 결과를 공식화한다.

2. 확정된 타당성 분석 결과를 이해관계자에게 배포하여 공유한다.

(1) 의사 결정자의 확인으로 공식화된 요구사항의 시스템화 타당성 분석 결과를 관
련 이해관계자에게 배포 공유함으로써 다음 단계로 진행될 수 있도록 한다.

수행 tip

- 기술의 복잡성, 검증 여부, 의존성 등 기술적 위험
분석 시 위험의 발생 가능성과 영향도를 동시에 고
려해야 한다.

학습 1	현행 시스템 분석하기
학습 2	요구사항 확인하기
학습 3	분석모델 확인하기

3-1. 분석모델 검증

학습 목표

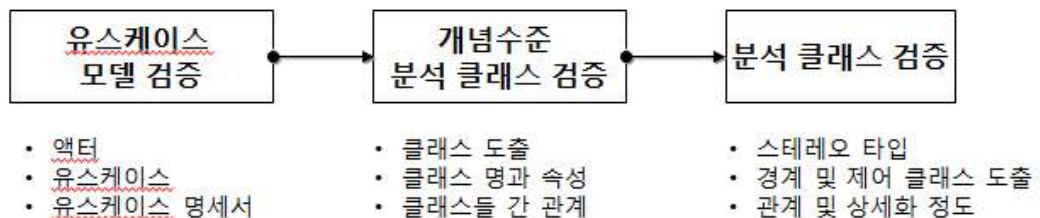
- 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석 모델에 대해서 확인할 수 있다.
- 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.

필요 지식 /

① 분석모델 검증

1. 검증 방법

한국정보화진흥원의 “정보시스템 감리지침: 시스템 개발사업 객체지향 컴포넌트 모델 V1.0”에서 사업유형이 시스템 개발, 감리 시점이 요구 분석, 감리 영역이 응용시스템인 경우 “유스케이스 모형 상세화 수준 및 적정성”과 “개념 수준의 분석클래스 도출”에 관한 점검항목을 기준으로 검증하고 감리 시점이 분석 설계, 감리 영역이 응용시스템인 경우 “유스케이스로부터 분석 클래스 도출 및 상세화”에 관한 점검항목을 기준으로 검증한다.



[그림 3-1] 분석모델 검증 절차

2. 유스케이스 모델 검증

한국정보화진흥원의 정보시스템 감리지침에서는 시스템 기능에 대한 유스케이스 모형 상세화 수준 및 적정성에 대하여 다음과 같은 사항을 점검하도록 하고 있다.

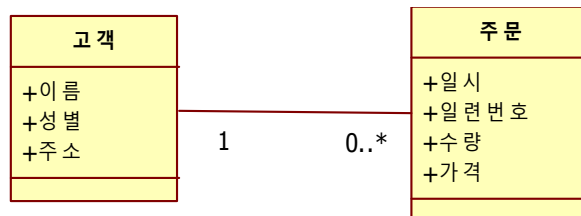
<표 3-1> 유스케이스 모델 검증

점검 대상	점검 내용
액터	<ul style="list-style-type: none"> - 기능 구현에 관계되는 액터가 모두 도출되었는가? - 액터 목록에서 액터명이 역할 중심으로 명명되었는가? - 요구사항 정의서, 요구사항 기술서에 외부/내부 액터가 모두 도출되었는가? - 액터 목록과 액터 명세서에 기록된 액터가 타당한지 확인
유스케이스	<ul style="list-style-type: none"> - 요구기능 구현에 필요한 유스케이스가 모두 도출되었는가? - 도출된 유스케이스를 논리적으로 연결하여 누락된 기능을 파악 - 도출된 유스케이스가 유스케이스 목록과 유스케이스 명세서에 반영되었는지 확인 - 도출된 유스케이스의 논리적인 합이 과업 범위와 일치하는지 비교 - 도출된 유스케이스들이 논리적으로 그룹화되었는지 확인(그룹화는 액터 기준, 연관 관계 기준, 동시성 기준이 가능) - 유스케이스 기능 범위가 다른 유스케이스 기능 범위와 중복되는지 확인
유스케이스 명세서	<ul style="list-style-type: none"> - 유스케이스 명세서 형식에 중요 항목이 누락되지 않았는지 확인 (사전 및 사후 조건, 주요 흐름, 서브 흐름, 예외 흐름 등) - 유스케이스의 주요 이벤트 흐름이 모두 도출되고 논리적으로 타당한지 확인 - 유스케이스를 구현하기 위하여 필요한 입출력 항목이 모두 도출되었는지 확인

3. 개념 수준의 분석 클래스 검증

시스템의 주요 도메인 개념을 분석 클래스로 도출하여 유스케이스 분석에 활용하므로, 개념 수준의 주요 분석 클래스를 적절히 도출하였는지, 관련 정보가 명확한지 점검해야 한다. 주요 점검 항목은 다음과 같다.

- 개별 유스케이스 단위로 작성하지 않고 시스템 전체를 대상으로 작성하였는가?
- 중요도가 높은 요구사항 또는 유스케이스에 필요한 엔터티 클래스가 도출되었는가?
- 도출된 클래스 이름과 설명이 이해관계자 간에 이견이 발생하지 않도록 명확한가?
- 클래스의 속성은 도출하였는가? 도출된 속성의 이름과 설명이 명확한가?
- 클래스들 간에 순환적 관계가 불필요하게 정의되어 있는가?
- 클래스 간의 관계에서 다중성(Multiplicity)이 정의되었는가?



[그림 3-2] 클래스 간 다중성(Multiplicity) 예

<표 3-2> 클래스 간 다중성

표기법	설명
0..1	0 또는 1개의 인스턴스 n..m은 n에서부터 m까지 범위
0..* 또는 *	0을 포함한 무한개의 인스턴스
1	1개의 인스턴스
1..*	1개 이상의 인스턴스

4. 분석 클래스 검증


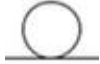

유스케이스마다 분석 클래스가 적절히 도출되었고, 제어 클래스의 도출 등이 충분하고 상세하게 도출되어 클래스의 역할, 클래스 간의 관계, 메시지 흐름 등을 확인할 수 있는지 검토한다.

(1) 유스케이스 실현(Realization)에 필요한 분석 클래스 도출 확인

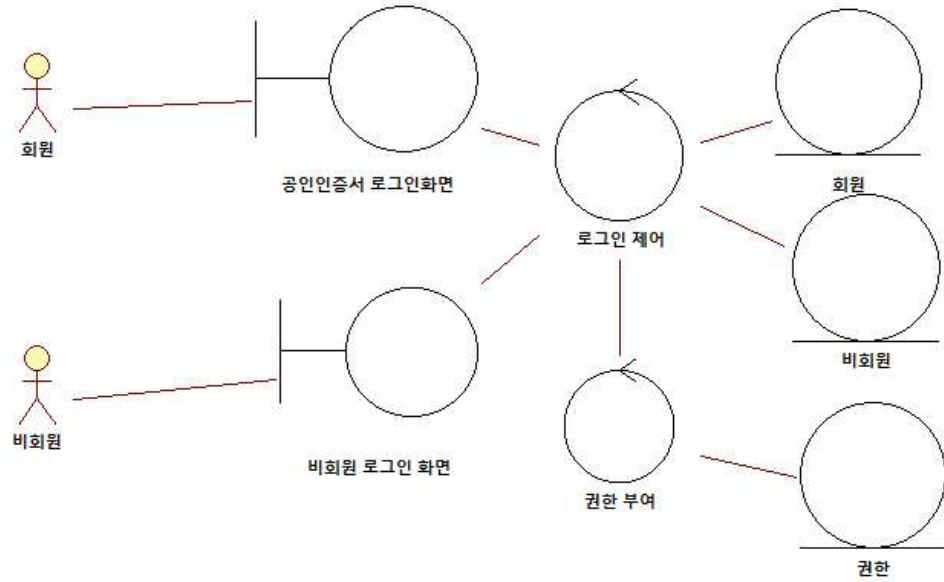
(가) 하나의 유스케이스를 실현하기 위하여 3개 이상의 클래스가 역할(Role) 기준으로 도출되어야 하며, 유스케이스 별로 실현에 필요한 클래스가 추적 가능해야 클래스 누락 여부를 확인할 수 있다.

(나) 유스케이스 별로 도출된 분석 클래스들이 역할(Role) 기준으로 경계(Boundary), 엔터티(Entity), 제어(Control) 클래스가 도출되어 스테레오 타입으로 표시되었는지 확인한다.

<표 3-3> 분석 클래스의 스테레오 타입

역할구분	스테레오 타입	내용
경계 Boundary	<<boundary>> 	시스템과 외부 액터와의 상호작용을 담당하는 클래스
엔터티 Entity	<<entity>> 	시스템이 유지해야 하는 정보를 관리하는 기능을 전담하는 클래스
제어 Control	<<control>> 	시스템이 제공하는 기능의 로직 및 제어를 담당하는 클래스

유스케이스 이벤트 흐름에 따라 다르지만 일반적으로 유스케이스 당 1개의 제어 클래스가 존재하고, 연결된 액터마다 1개의 경계 클래스가 존재하는지 확인한다.



[그림 3-3] 분석 클래스 다이어그램 예시

(2) 경계(Boundary)와 제어(Control) 클래스의 도출 여부 및 상세화 정도 확인

유스케이스 실현에 필요한 분석 클래스들이 도출되었는지 확인하기 위하여, 유스케이스 단위로 분석 클래스를 확인한다.

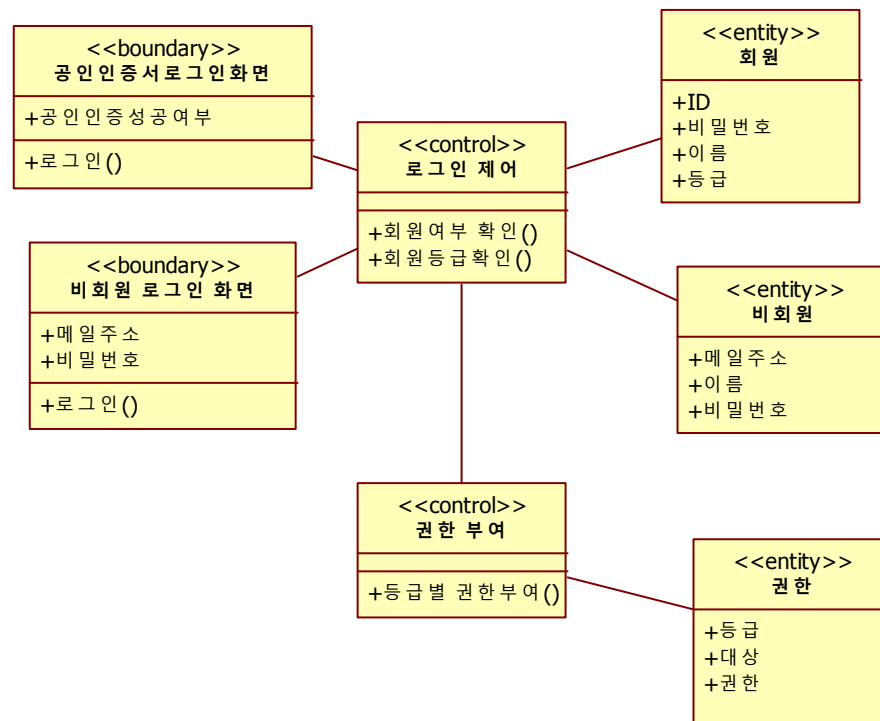
<표 3-4> 경계 및 제어 클래스 도출 및 상세화 정도 확인

역할 구분	검토 사항
경계	<ul style="list-style-type: none"> - 유스케이스와 연결된 액터가 있고, 액터의 유형이 시스템 또는 장비인 경우, 해당 액터를 위한 경계 클래스가 도출되었는지 확인하고, 유스케이스의 이벤트 흐름을 참조하여 관련 기능을 처리하기 위한 연산이 도출되었는지 확인한다. - 유스케이스 명세서의 이벤트 흐름을 확인하여, 유스케이스에서 필요한 UI를 위한 경계 클래스가 도출되었는지 확인한다. - UI를 위한 경계 클래스인 경우, 사용자에게 제공할 항목이 속성으로 도출되었는지 확인하고, 화면, 보고서 상의 데이터 타입, 길이가 경계 클래스 속성 정의와 일치하는지 확인한다.
제어	<ul style="list-style-type: none"> - 유스케이스 별로 제어 클래스가 1개 이상 도출되었는지 확인한다. - 제어 클래스의 연산에 대응하는 엔터티 클래스가 있는지 확인한다. - 유스케이스 명세서 기술된 이벤트 흐름을 처리하기 위한 연산이 제어 클래스에 정의되어 있는지 확인한다.

(3) 클래스 간의 관계, 클래스 정보의 상세화 정도 확인

<표 3-5> 분석 클래스 간 관계 및 상세화 정도 확인

역할 구분	검토 사항
관계	<ul style="list-style-type: none"> - 유스케이스 명세서를 바탕으로 각 클래스 사이의 관계를 정의하였는지 확인한다. - 관계의 다중성이 정확하고 모순이 없는지 확인한다. - 2개의 클래스 간에 1개 이상의 관계가 존재하면, 관계 명 또는 역할 명이 정의되었는지 확인한다.
연산 및 속성 상세화	<ul style="list-style-type: none"> - 유스케이스 명세서를 바탕으로 클래스의 속성 및 연산이 도출되었는지 확인한다. - 도출된 연산의 매개 변수(명, 타입, 길이)와 리턴 타입이 정의되었는지 확인한다. - 도출된 클래스의 속성(명, 타입, 길이)이 이해관계자 간에 이견이 없도록 명확하게 정의되었는지 확인한다. - 경계 클래스의 속성과 화면/보고서의 항목, 엔터티 클래스의 속성 정보가 일관성을 가지는지 확인한다.



[그림 3-4] 분석 클래스 다이어그램 예시

수행 내용 / 분석모델 검증하기

재료·자료

- 요구사항 목록, 요구사항 정의서, 유스케이스 다이어그램, 유스케이스 명세서, 액터 목록, 유스케이스 목록, 개념 수준 분석 클래스 다이어그램 및 명세서, 분석 클래스 다이어그램 및 명세서, 분석 클래스 목록

기기(장비·공구)

- 컴퓨터, 프린터, 인터넷, 문서 작성 프로그램, UML 저작 프로그램

안전·유의사항

- 실습 시 지도 교사의 지시를 따른다.

수행 순서

- ① 분석모델까지 요구사항 추적표를 작성하고 검토 의견 컬럼을 추가한다.

〈표 3-6〉 요구사항 추적표 - 검토 의견 추가 표 형식

요구사항			유스케이스				개념 수준 분석 클래스		분석 클래스		
ID	이름	ID	이름	관련 액터	명세 서 명	검토 의견	클래스명	검토 의견	클래스명	유형	검토 의견

- ② 작성된 요구사항 추적표에 검토 의견을 작성한다.

- 요구사항 목록을 참조하여 요구사항 ID와 요구사항 명을 입력한다.
- 유스케이스 모델 검토 의견을 작성한다.

요구사항 정의서, 액터 목록, 유스케이스 목록, 유스케이스 다이어그램, 유스케이스 명세서를 참조하여 유스케이스 ID, 유스케이스 이름, 유스케이스 명세서 명을 입력하고, 관련된 액터를 찾아서 입력한다. 지식에서 기술된 유스케이스 모델 검증 내용을 참조하여 액터, 유스케이스, 유스케이스 명세서 관련 검토 의견을 작성한다.

<표 3-7> 유스케이스 모델 검토 의견 작성 예시

항목	내용
요구사항 ID	Req-001
요구사항 이름	회원은 공인 인증서로 로그인할 수 있어야 한다.
유스케이스 ID	UC-001
유스케이스 이름	로그인한다
관련 액터	사용자, 회원, 비회원, 관리자
명세서명	로그인한다
검토 의견	<ul style="list-style-type: none"> • 기능 구현에 관련된 모든 액터가 도출되었으며, 액터 명이 역할 중심으로 명명되었음 • 요구 기능에 필요한 유스케이스는 회원 공인 인증서로 인증하기, 권한 부여하기가 있음 • 도출된 유스케이스들이 회원 공인 인증서 로그인 과업 범위와 일치하고 있음 • 유스케이스 명세서에 중요 항목들이 포함되어 있으며, 유스케이스 구현에 필요한 입출력 항목들이 모두 도출되었음

유스케이스 모델 검토 의견을 작성할 때 <표 3-6>과 같이 엑셀(Excel)에 작성하는 것이 일반적이나, 문서 편집기 툴에서는 내용을 입력하기 어렵기 때문에 위와 같이 형태를 변경하여 예시를 제시하였다.

해당 요구사항을 구현하기 위해서는 추가적으로 회원 공인 인증서로 인증하기, 권한 부여하기 유스케이스를 검토해야 한다.

3. 개념 수준 분석 클래스 모델 검토 의견을 작성한다.

유스케이스 다이어그램 및 명세서, 개념 수준 분석 클래스 다이어그램 및 명세서를 참조하여 유스케이스에 대응되는 클래스 명을 입력하고, 필요 지식에서 기술된 개념 수준의 분석 클래스 검증 내용을 참조하여 검토 의견을 작성한다.

4. 분석 클래스 모델 검토 의견을 작성한다.

유스케이스 목록, 유스케이스 다이어그램, 유스케이스 명세서, 개념 수준 분석 클래스 다이어그램 및 명세서, 분석 클래스 목록, 분석 클래스 다이어그램 및 명세서를 참조하여, 대응되는 분석 클래스 명, 분석 클래스의 유형(경계, 제어, 엔터티)을 입력하고, 필요 지식에서 기술된 분석 클래스 검증 내용을 참조하여 분석 클래스 도출 유형, 관계, 연산 및 속성 상세화 관련 검토 의견을 작성한다.

<표 3-8> 분석 클래스 모델 검토 의견 작성 예시

항목	내용
요구사항 ID	Req-001
요구사항 이름	회원은 공인 인증서로 로그인할 수 있어야 한다.
클래스 명	공인 인증서 로그인 화면
유형	경계
검토 의견	<ul style="list-style-type: none"> • 유스케이스와 연결된 액터가 존재하고, 해당 액터를 위한 경계 클래스가 도출됨 • 유스케이스 명세서의 이벤트 흐름에 적합한 U를 위한 경계 클래스가 도출됨 • 관계의 다중성이 표현되어 있으며, 클래스의 속성 및 연산이 유스케이스 명세서를 기반으로 도출됨

분석 클래스 모델 검토 의견을 작성할 때 <표 3-6>과 같이 엑셀(Excel)에 작성하는 것이 일반적이나, 문서 편집기 툴에서는 내용을 입력하기 어렵기 때문에 위와 같이 형태를 변경하여 예시를 제시하였다.

③ 요구사항 추적표에서 요구사항에 대한 검토 의견을 정제한다.

1. 누락된 유스케이스 모델/개념 수준 분석 클래스/분석 클래스가 존재하는 경우, 검토 의견에 “작성필요” 라고 입력한다.
2. 단, 비기능 요구사항도 같이 관리하는 경우에는 검토 의견에 “비기능 요구사항, 작성 불필요” 라고 입력한다.

수행 tip

- 개념 수준의 분석 클래스 모델링을 생략하거나, 관련된 산출물의 일부(목록 등)를 제외하고 실습을 수행할 수 있다.

3-2. 분석모델의 시스템화 타당성 분석

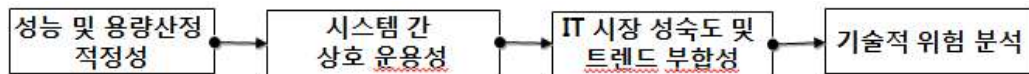
학습 목표

- 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다.

필요 지식 /

① 분석모델의 기술적 타당성 검토

유스케이스 모델의 개별 유스케이스에 대한 분석모델을 작성한 이후, 해당 분석모델로 시스템을 개발하는 경우에 어떠한 영향을 미치는지 필요한 자원, 상호 운용성, 시장 성숙도, 기술적 위험 분석 측면에서 타당성을 조사한다.



[그림 3-5] 분석모델의 기술적 타당성 검증 절차

<표 3-9> 분석모델의 기술적 타당성 검토

검토 분야	검토 내용
성능 및 용량	<ul style="list-style-type: none"> - 요구사항을 만족시키기 위한 분석모델에 따라 시스템을 구현할 때 요구되는 시스템의 자원을 식별한다. - 분석 클래스에서 불필요하고 지나치게 많고 속성들을 포함시키게 되면 객체 생성 시 시스템의 메모리 자원을 많이 요구하게 되며, 이로 인한 JVM에서 과도한 가비지 컬렉션(Garbage Collection)이 발생하여 전체 시스템의 성능 저하가 빈번히 발생한다.
시스템 간 상호 운용성	<ul style="list-style-type: none"> - 분석모델을 이용하여 보다 구체적으로, 시스템 간 상호 정보 및 서비스를 교환 가능한지 검토한다. - 분석모델에서 정의한 구체적인 정보의 존재 여부, 생성 가능성, 교환 방식 지원 등에 대해서 확인한다.
시장 성숙도 및 트렌드 부합성	<ul style="list-style-type: none"> - 분석모델이 과거의 문제를 해결하고 많이 사용되는 트렌드에 부합하는지 확인한다. - 예를 들어, 시스템에서 중요하고 빈번하게 사용되는 클래스를 Spring의 프로토타입 빈(Prototype Bean)으로 사용할 것을 가정하고 분석모델이 작성되지 않았는지 검토한다.
기술적 위험 분석	<ul style="list-style-type: none"> - 분석모델이 시스템의 기술 구조, 프레임워크, 사용되는 하드웨어 및 소프트웨어와 부합되는지 확인한다. - 분석모델이 검증되지 않은 기술의 사용을 가정으로 하고 있어 추가적인 비용 발생 가능성이 있는지 확인한다. - 분석모델을 구현하기 위하여 특정 업체 기술, 특허, 라이선스에 의존해야 하는지 확인한다.

수행 내용 / 분석모델의 시스템화 타당성 분석하기

재료·자료

- 현행 시스템 분석서, 사용자 면담 기록, 요구사항 목록, 요구사항 정의서, 유스케이스 다이어그램, 유스케이스 명세서, 유스케이스 목록, 개념 수준 분석 클래스 다이어그램 및 명세서, 분석 클래스 다이어그램 및 명세서, 분석 클래스 목록

기기(장비·공구)

- 컴퓨터, 프린터, 인터넷, 문서 작성 프로그램, 엑셀(Excel) 프로그램, UML 저작 프로그램

안전·유의사항

- 실습 시 지도 교사의 지시를 따른다.

수행 순서

- ① 분석모델까지 요구사항 추적표를 작성하고, 타당성 검토 의견 컬럼을 추가한다.

<표 3-10> 요구사항 추적표 - 타당성 검토 의견 추가

요구사항			유스케이스			개념 수준 분석 클래스			분석 클래스		
ID	이름	ID	이름	관련 액터	명세 서 명	검토 의견	클래스 명	검토 의견	클래스 명	유형	검토 의견

- ② 작성된 요구사항 추적표에 타당성 검토 의견을 작성한다.

1. 타당성 검토 의견을 제외한 나머지 속성들은 분석모델 검증 수행 내용의 작성 절차와 동일하다.
2. 유스케이스 모델, 개념 수준 분석 클래스 모델, 분석 클래스 모델의 기술적 타당성 검토를 위하여 필요 지식에 명시된 바와 같이 성능 및 용량, 시스템 간 상호 운용성, 시장 성숙도 및 트렌드 부합성, 기술적 위험 분석을 참조하여 검토 의견을 작성한다. 각 모델별로 다음 사항을 특히 유의해야 한다.

(1) 유스케이스 모델 타당성 검토

요구사항을 만족시키기 위한 유스케이스 모델을 검토하여 관련 액터들이 지나치게 많은지, 이벤트 흐름과 예외 흐름이 지나치게 복잡한지 등을 검토해야 한다. 이는 요구사항이 세분화되어 있지 않거나, 해당 요구사항을 만족시키기 위한 기술적 복잡도가 증가한다는 의미로 추가적인 비용 발생 가능성이 크다.

(2) 개념 수준 분석 클래스 모델 타당성 검토

중요도가 높은 요구사항을 위주로 작성된 개념 수준 분석 클래스 모델을 검토할 때는 너무 많은 개념 수준 분석 클래스 모델들이 도출되지 않았는지 검토한다. 이는 요구사항을 만족시키기 위한 전체 시스템의 복잡도가 높다는 의미이며, 비용 추가 가능성을 내포하고 있다.

(3) 분석 클래스 모델 타당성 검토

요구사항을 만족시키기 위한 경계/제어/엔터티 클래스들이 너무 많은지, 관계가 너무 복잡한지, 관리해야 할 속성들이 지나치게 많은지 검토한다. 이러한 분석 클래스는 복잡성을 증가시켜 전체적인 성능을 저하시키고 개발 생산성을 저하시킬 가능성이 크다.

일반적으로 엑셀(Excel)을 이용하여 검토 결과를 작성할 때 <표 3-10>과 같은 형태를 사용하나, 문서 편집기 툴에서는 내용을 입력하기 어렵기 때문에 <표 3-11>과 같이 형태를 변경하여 분석모델에 대한 타당성 검토 결과를 제시한다.

<표 3-11> 분석모델의 분석 클래스의 기술적 타당성 검토 예시

항목	내용
요구사항 ID	Req-001
요구사항 이름	회원은 공인 인증서로 로그인할 수 있어야 한다.
분석 클래스 명	공인 인증서 로그인 화면
분석 클래스 유형	경계
검토 의견	성능 및 용량: 회원별로 공인 인증서 로그인 화면 클래스의 인스턴스가 생성되지 않도록 화면에서 전달되는 정보들을 별도의 VO(Value Object) 형태의 객체에 저장하거나 해시 맵(Hash Map) 형태를 사용하도록 수정이 필요하다. 사용자별로 이 클래스의 인스턴스를 생성하게 되는 경우 메모리 상에 많은 객체들이 생성하게 되어 빈번한 GC(Garbage Collection)가 발생할 확률이 증가한다. 시스템 간 상호 운용성: 공인 인증서 로그인 화면 클래스에서 로그인 제어 클래스로 정보를 전달하거나 엔터티 클래스로 정보를 전달할 때 전달되는 데이터형식의 일치가 필요하다. 또 화면에서 전달되는 데이터가 공인 인증서 로그인 화면 클래스에서 적절히 사용할 수 있는 데이터 형식으로 변환이 필요하다.

항목	내용
	<p>시장 성숙도 및 트렌드 부합성</p> <p>MVC 패턴에 적합하게 경계/제어/엔터티 클래스들이 도출되고 있다. 하지만, 경계 클래스의 데이터를 엔터티 클래스들로 넘겨주어 저장하는 방식에 대해서는 기술적 위험에 대하여 검토가 필요하다.</p> <p>기술적 위험 분석</p> <p>경계 클래스의 데이터가 제어 클래스를 거쳐 엔터티 클래스로 전달된 경우 이를 어떤 방식으로 데이터에 저장하느냐에 대한 다양한 ORM(Object-Relational Mapping) 기법들이 존재한다. 이 중에서, 일반적으로 많이 사용되는 MyBatis 프레임워크를 고려할 필요가 있다.</p>

③ 타당성 분석 결과를 관련 이해관계자가 검증한다.

1. 타당성 분석 결과를 관련 이해관계자에게 배포하여 사전 검토를 요청한다.
타당성 분석 결과 검증을 위한 회의 이전에 관련 이해관계자가 사전 검토를 하도록 함으로써 검증 작업이 효과적 효율적으로 진행될 수 있도록 만들기 위한 것이다.
2. 관련 이해관계자가 모여 분석모델 타당성 분석 결과를 검증한다.
관련 이해관계자 간 이견이 있을 수 있으므로 모든 관련 이해관계자가 검증 회의에 참여하는 것이 중요하다.
3. 타당성 분석 결과에 이견이 있는 경우 프로젝트 관리자(Project Manager)의 중재 하에 합의를 도출한다.
타당성 분석결과가 관련 이해관계자 간 이견이 있는 상태로 남아 있지 않도록 프로젝트 관리자의 중재 하에 타당성 분석 결과의 이견을 해소하도록 한다.

④ 관련 이해관계자 검증을 거친 타당성 분석 결과를 확인하고 배포 및 공유한다.

1. 관련 이해관계자 검증을 거친 타당성 분석 결과를 의사 결정자가 확인한다.
의사 결정자의 확인을 통해 타당성 분석 결과의 검증 작업 결과를 공식화한다.
2. 의사 결정자가 확인한 타당성 분석 결과를 이해관계자에게 배포하여 공유한다.
의사 결정자의 확인으로 공식화된 타당성 분석 결과를 배포 공유함으로써 다음 단계로 진행될 수 있도록 한다.

수행 tip

- 분석모델의 기술적 타당성 검토는 기술적 위험 분석을 위주로 수행되는 것이 바람직하다.