

중단원	소단원	번호	문제	키워드	답
		1	어떤 하위 시스템으로 구성되어 있고, 제공 기능 및 연계 정보는 무엇이며, 어떤 기술 요소를 사용하는지를 파악하는 활동이다	현행 시스템 파악	현행 시스템 파악
		2	현행 시스템 파악 절차 작성하시오	현행 시스템 파악	<ul style="list-style-type: none"> <li>* 1단계) 현행 시스템 구성/기능/인터페이스 파악 <ul style="list-style-type: none"> <li>- 시스템 구성 현황 파악</li> <li>- 시스템 기능 파악</li> <li>- 시스템 인터페이스 파악</li> </ul> </li> <li>* 2단계) 현행 시스템 아키텍처 및 소프트웨어 구성 파악 <ul style="list-style-type: none"> <li>- 아키텍처 파악</li> <li>- 소프트웨어 구성 파악</li> </ul> </li> <li>* 3단계) 하드웨어 및 네트워크 구성 파악 <ul style="list-style-type: none"> <li>- 시스템 하드웨어 현황 파악</li> <li>- 네트워크 구성 파악</li> </ul> </li> </ul>
		3	현행 시스템 구성 현황 파악 내용 설명	시스템 구성 현황 파악	<ul style="list-style-type: none"> <li>* 정의 <ul style="list-style-type: none"> <li>- 조직의 주요 업무 처리하는 기간 업무와 이를 지원하는 지원 업무로 구분하여 기술한 현황</li> </ul> </li> <li>* 작성 시 고려사항 <ul style="list-style-type: none"> <li>- 각 업무에 속하는 단위 업무 정보시스템들의 명칭, 주요 기능들을 명시함으로써 조직 내 존재하는 모든 정보시스템의 현황 파악이 가능하도록 함</li> </ul> </li> </ul>
		4	현행 시스템 파악 절차 중 인터페이스 파악의 시 고려 사항	인터페이스 파악	데이터 형식, 통신규약, 연계유형 파악
		5	현행 시스템 파악 절차 중 소프트웨어 구성 파악의 시 고려 사항	소프트웨어 파악	제품명, 용도, 라이선스 적용방식, 라이언스 수
		6	소프트웨어 아키텍처 개념에 대해 서술하시오	소프트웨어 아키텍처	여러가지 소프트웨어 구성요소와 그 구성요소가 가진 특성 중 외부에 드러나는 특성, 그리고 구성요소 간의 관계를 표현하는 시스템 구조이다
		7	소프트웨어 아키텍처 프레임워크 개념	소프트웨어 아키텍처 프레임워크	소프트웨어 집약적인 시스템에서 아키텍처가 표현해야 하는 내용 및 이들 간의 관계를 제공하는 아키텍처 기술 표준이다

	1. 현행 시스템 파악	8	소프트웨어 아키텍처 프레임워크 구성요소	소프트웨어 아키텍처 프레임워크	<ul style="list-style-type: none"> <li>* 아키텍처 명세서 <ul style="list-style-type: none"> <li>- 아키텍처 기록하기 위한 <u>산출물</u></li> <li>- 이해관계자들의 시스템에 대한 관심을 관점에 맞춰 작성한 뷰</li> <li>- 개별 뷰, 뷰 개괄 문서, 인터페이스 명세 등이 있음</li> </ul> </li> <li>* 이해관계자 <ul style="list-style-type: none"> <li>- 시스템 개발에 관련된 <u>모든 사람과 조직</u></li> <li>- 고객, 최종사용자, 개발자, 프로젝트 관리자, 유지보수자, 마케팅 담당자 등 모두 포함</li> </ul> </li> <li>* 관심사 <ul style="list-style-type: none"> <li>- 시스템에 대해 이해관계자들의 서로 다른 의견과 목표</li> <li>- 사용자 입장 / 유지보수자 입장 / 개발자 입장</li> </ul> </li> <li>* 관점 <ul style="list-style-type: none"> <li>- <u>개별 뷰</u>를 개발할 때 토대가 되는 <u>패턴</u>이나 <u>양식</u></li> <li>- 이해관계자들이 서로 다른 역할이나 책임으로 시스템이나 산출물들에 대해 보고 싶은 관점</li> </ul> </li> <li>* 뷰 <ul style="list-style-type: none"> <li>- 서로 관련 있는 관심사들의 집합이라는 관점에서 전체 시스템 표현</li> <li>- 시스템에 대한 아키텍처 설명에는 하나 이상의 뷰로 구성</li> </ul> </li> <li>* 근거 <ul style="list-style-type: none"> <li>- 아키텍처 결정 근거</li> </ul> </li> </ul>
				소프트웨어 아키텍처 4+1 뷰 설명	소프트웨어 아키텍처 4+1뷰 - 유논프구배

01. 현행 시스템 분석

10	소프트웨어 아키텍처 4+1뷰 설명	소프트웨어 아키텍처 4+1뷰 - 유논프구배	<ul style="list-style-type: none"> <li>* 유스케이스 뷰                     <ul style="list-style-type: none"> <li>- 아키텍처 도출하고 설계하는 <u>작업 주도하는 뷰</u></li> <li>- 다른 뷰 검증하는 데 사용</li> </ul> </li> <li>* 논리 뷰                     <ul style="list-style-type: none"> <li>- <u>설계 모델의 추상화</u>이며, 주요 설계 패키지와 서브 시스템, 클래스 식별하는 뷰</li> <li>- <u>시스템 기능적인 요구사항 지원</u></li> <li>- 클래스와 이들 간 관계에 대한 집합 보여주는 <u>클래스 다이어그램으로 표현</u></li> </ul> </li> <li>* 프로세스 뷰                     <ul style="list-style-type: none"> <li>- 런타임 시의 시스템 <u>태스크, 스레드, 프로세스</u>와 이들 사이의 <u>상호작용 등의 관계</u> 표현하는 뷰</li> <li>- 성능이나 가용성과 같은 시스템의 <u>비기능적인 요구사항</u> 고려</li> </ul> </li> <li>* 구현 뷰                     <ul style="list-style-type: none"> <li>- 개발 환경 안에서 정적인 <u>소프트웨어 모듈의 구성을 표현</u>하는 뷰</li> <li>- 정적인 소프트웨어 모듈 : 소스 코드, 데이터 파일, 컴포넌트, 실행 파일 등</li> <li>- <u>개발자 관점에서</u> 소프트웨어 구현과 관리적인 측면을 <u>컴포넌트 다이어그램으로 표현</u></li> <li>- <u>컴포넌트 뷰</u>라고도 함</li> </ul> </li> <li>* 배포 뷰                     <ul style="list-style-type: none"> <li>- <u>물리적인 노드의 구성과 상호 연결 관계</u>를 배포 다이어그램으로 표현하는 뷰</li> <li>- 다양한 실행 파일과 다른 런타임 컴포넌트가 해당 플랫폼 또는 컴퓨팅 <u>노드</u>에 어떻게 매핑되는가를 보여주며, 가용성, 신뢰성, 성능, 확장성 등의 시스템의 비기능적인 요구사항 고려</li> </ul> </li> </ul>
11	현행 시스템 분석서 작성 및 검토 절차 나열	현행 시스템 분석서	<ol style="list-style-type: none"> <li>1. 현행 시스템 관련 자료 <u>수집</u> 단계</li> <li>2. 자료 <u>분석</u> 단계</li> <li>3. 산출물 <u>작성</u> 단계</li> <li>4. 산출물에 대한 <u>검토</u> 수행 단계</li> </ol>
12	현행 시스템 분석 시 현행 시스템 아키텍처 구성도 자료에 대한 분석 대상 2가지 작성	현행 시스템 분석	<ul style="list-style-type: none"> <li>- 운영체제의 종류</li> <li>- 적용 프레임워크</li> <li>- 계층별 적용 기술</li> <li>- 데이터 연동 방식</li> <li>- 데이터 저장소</li> <li>- 외부 시스템 연계 방식</li> </ul>
13	<ul style="list-style-type: none"> <li>- 송/수신 시스템</li> <li>- 연계 방식 및 연계 주기</li> <li>- 연동 데이터 형식</li> <li>- 주요 연동 데이터 : 산출물에 대한 검토 수행 단계</li> </ul> <p>어떤 취득 자료를 통해 위 분석 대상 자료를 도출한 것인가</p>	현행 시스템 분석	인터페이스 현황
14	인터페이스 현황 자료에 대한 분석 대상	현행 시스템 분석	<ul style="list-style-type: none"> <li>- 송/수신 시스템</li> <li>- 연계 방식 및 연계 주기</li> <li>- 연동 데이터 형식</li> <li>- 주요 연동 데이터</li> </ul>

	15	현행 시스템 분석 결과를 기반으로 작성한 산출물의 종류	현행 시스템 분석	1. 정보시스템 구성 현황 2. 정보시스템 기능 구성도 3. 인터페이스 현황 4. 현행 시스템 아키텍처 구성도 5. 소프트웨어 구성도 6. 하드웨어 구성도 7. 네트워크 구성도
2. 개발 기술 환경 정의	1	운영체제 현행 시스템 분석 시 성능 측면 고려사항 간략히 서술	운영체제 현행 시스템 분석	* 품질 측면 - 신뢰도 - 성능 * 대규모 및 대량 파일 작업(Batch Job) 처리 가능 여부 * 지원 가능한 메모리 크기 고려  * 지원 측면 - 기술 지원 - 주변 기기 - 구축 비용
	2	운영체제 현행 시스템 분석 시 고려 사항	운영체제 현행 시스템 분석	* 품질 측면 : 신뢰도, 성능 * 지원 측면 : 기술 지원, 주변 기기, 구축 비용
	3	OSI 7계층 설명	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	응용 계층 - 표현 계층 - 세션 계층 - 전송 계층 - 네트워크 계층 - 데이터 링크 계층 - 물리 계층
	4	응용 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 사용자와 네트워크 간 응용 서비스 연결, 데이터 생성 * 프로토콜 : HTTP, FTP * 전송 단위 : 데이터(Data)
	5	표현 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 데이터 형식 설정과 부호 교환, 암호/복호화 * 프로토콜 : JPEG, MPEG * 전송 단위 : 데이터(Data)
	6	세션 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 연결 접속 및 동기 제어 * 프로토콜 : SSH, TLS * 전송 단위 : 데이터(Data)
	7	전송 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 신뢰성 있는 통신 보장 * 데이터 분할과 재조립, 흐름제어, 오류제어 등을 담당 * 오류제어, 혼잡제어 등 * 프로토콜 : TCP, UDP * 전송 단위 : 세그먼트(Segment)
	8	네트워크 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 단말 간 데이터 전송을 위한 최적화된 경로 제공 * 프로토콜 : IP, ICMP * 전송 단위 : 패킷(Packet)
	9	데이터 링크 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 인접 시스템 간 데이터 전송, 전송오류 제어 * 동기화, 오류 제어, 흐름 제어 등의 전송 에러를 제어 * 오류 검출 / 재전송 등 * 프로토콜 : 이더넷 * 전송 단위 : 프레임(Frame)
	10	물리 계층	네트워크 현행 시스템 분석 : OSI 7계층 (Layer)	* 0과 1의 비트 정보를 회선에 보내기 위한 전기적 신호 변환 * 프로토콜 : RS-232C * 전송 단위 : 비트(Bit)

		11	DMBS 현행 시스템 분석 시 고려사항	DMBS 현행 시스템 분석	<ul style="list-style-type: none"> <li>* 성능 측면 <ul style="list-style-type: none"> <li>- 가용성 <ul style="list-style-type: none"> <li>* 장기간 시스템을 운영할 때 장애 발생 가능성</li> <li>* 백업 및 복구 편의성</li> <li>* DMBS 이중화 및 복제 지원 여부</li> </ul> </li> <li>- 성능 <ul style="list-style-type: none"> <li>* 대규모 데이터 처리 성능</li> <li>* 대량 거래 처리 성능</li> <li>* 다양한 튜닝 옵션 지원 여부</li> <li>* 비용 기반 최적화 지원 및 설정의 최소화 지원 여부</li> </ul> </li> <li>- 상호 호환성 <ul style="list-style-type: none"> <li>* 설치 가능한 운영체제 종류</li> <li>* 다양한 운영체제에서 지원되는 JDBC, ODBC</li> </ul> </li> </ul> </li> <li>* 지원 측면 <ul style="list-style-type: none"> <li>- 기술 지원</li> <li>- 구축 비용</li> </ul> </li> </ul>
		12	웹 애플리케이션 서버(WAS)의 개념	미들웨어 현행 시스템 분석	<ol style="list-style-type: none"> <li>1. 애플리케이션 동작 환경 제공</li> <li>2. 안정적인 트랜잭션 처리와 관리</li> <li>3. 다른 이기종 시스템과 애플리케이션 연동 지원 서버</li> </ol> <p>서버계층에서 애플리케이션이 동작할 수 있는 환경을 제공하고 안정적인 트랜잭션 처리와 관리, 다른 이기종 시스템과의 애플리케이션 연동을 지원하는 서버이다</p>
		13	기술 환경 정의 위한 기초 자료 조사 항목 사례 용어 * 온라인 트랜잭션 처리(OLTP) 시스템	개발 기술 환경 요구사항 파악	<ul style="list-style-type: none"> <li>- 시스템 구축 형태</li> <li>- 사용자 수</li> <li>- 트랜잭션 수</li> </ul>
		14	기술 환경 정의 위한 기초 자료 조사 항목 사례 용어 * 웹/웹 애플리케이션 서버(WEB/WAS) 시스템	개발 기술 환경 요구사항 파악	<ul style="list-style-type: none"> <li>- 시스템 용도 및 서비스 형태</li> <li>- 시스템 구성 형태 (1계층, 2계층, 3계층)</li> <li>- 접속자 수</li> </ul>
		15	$CPU(tpmC \text{ 단위}) = \text{분당 트랜잭션 수} * \text{기본 tpmC 보정} * \dots$ 위 식의 대상이 되는 서버의 유형	CPU 용량 산적	<p>OLTP / 배치 / 데이터 베이스 서버</p> <p>* 작업 부하를 위해 TPC-C 기준을 참조하여 tpmC를 사용</p>
		16	$CPU(OPS \text{ 단위}) = \text{동시 사용자 수} * \text{기본 OPS 보정} * \dots$ 위 식의 대상이 되는 서버의 유형	CPU 용량 산적	<p>WEB/WAS 서버</p>
				1	<ol style="list-style-type: none"> <li>1. ( 기능적 요구사항 ) - 시스템이 제공하는 기능, 서비스에 대한 요구사항</li> <li>2. ( 비기능적 요구사항 ) - 시스템이 수행하는 기능 이외의 사항, 시스템 구축 제약사항에 관한 요구사항</li> </ol>
2	요구사항 개발 프로세스 과정 순서대로 나열			요구사항 개발 프로세스	<ol style="list-style-type: none"> <li>1. 요구사항 도출</li> <li>2. 요구사항 분석</li> <li>3. 요구사항 명세</li> <li>4. 요구사항 확인</li> </ol>

1.요구사항

3	요구사항 도출 기법	요구사항 도출	<ul style="list-style-type: none"> <li>* 인터뷰                     <ul style="list-style-type: none"> <li>- 1:1 관계에서 시스템에 대한 요구사항 추출</li> <li>- 철저한 사전 준비 작업 필요</li> </ul> </li> <li>* 설문조사                     <ul style="list-style-type: none"> <li>- 설문지 또는 여론조사 등을 이용해 간접적으로 정보 수집</li> <li>- 개발될 시스템의 사용자가 다수일 때 의견 수렴 용이</li> </ul> </li> <li>* 브레인스토밍                     <ul style="list-style-type: none"> <li>- 말 꺼내기 쉬운 분위기에서, 회의 참석자들이 내놓은 아이디어 비판 없이 수용할 수 있도록 하는 회의</li> </ul> </li> <li>* 워크숍                     <ul style="list-style-type: none"> <li>- 단기간 집중적인 노력 통해 다양하고 전문적인 정보 획득하고 공유</li> <li>- 프로젝트에 참여하는 모든 핵심 인물의 참여가 필요</li> <li>- 참석자들은 해당 전문 영역별로 팀 협력이 필요</li> <li>- 사전 준비가 요구됨</li> </ul> </li> </ul>
	요구사항 분석 기법	요구사항 분석	<ul style="list-style-type: none"> <li>* 자료 흐름 지향 분석 기법                     <ul style="list-style-type: none"> <li>- 데이터 흐름(DFD : Data Flow Diagram)으로부터 소프트웨어 구조를 유도하는 방법</li> </ul> </li> <li>* 객체지향 분석                     <ul style="list-style-type: none"> <li>- 시스템의 기능과 데이터를 함께 분석</li> <li>- UML로 표준화</li> </ul> </li> </ul>
5	요구사항 분석 종류	요구사항 분석	<ol style="list-style-type: none"> <li>1. 요구사항 분류</li> <li>2. 개념 모델링</li> <li>3. 요구사항 할당</li> <li>4. 요구사항 협상</li> <li>5. 정형분석</li> </ol>
6	요구사항 분류에 대해 서술하시오	요구사항 분석	<ul style="list-style-type: none"> <li>* 요구사항이 기능인지 비기능인지 확인</li> <li>* 요구사항이 소프트웨어에 미치는 영향 범위 파악</li> <li>* 요구사항이 소프트웨어 생명주기 동안 변경이 발생하는지 확인</li> </ul>
7	요구사항 할당에 대해 서술하시오	요구사항 분석	<ul style="list-style-type: none"> <li>* 요구사항을 만족시키기 위한 아키텍처 구성요소를 식별하는 활동</li> <li>* 다른 구성요소와 어떻게 상호작용 하는지 분석을 통해 추가적인 요구사항을 발견 가능함</li> </ul>
8	요구사항 협상에 대해 서술하시오	요구사항 분석	<ul style="list-style-type: none"> <li>* 요구사항 협상은 두 명의 이해관계자가 서로 상충되는 내용을 요구하는 경우, 어느 한 쪽을 지지하기 보다는 적절한 지점에서 합의하기 위한 기법이다</li> </ul>
9	요구사항 명세 상세 내용	요구사항 명세	<ul style="list-style-type: none"> <li>* 체계적으로 검토, 평가, 승인될 수 있는 문서 작성하는 단계</li> <li>* 이 단계에서는 시스템 정의, 시스템 요구사항, 소프트웨어 요구사항을 작성한다</li> </ul>
10	<ul style="list-style-type: none"> <li>* 분석가가 요구사항을 이해했는지 ( ① )이 필요</li> <li>* 요구사항 문서가 회사의 표준에 적합하고 이해 가능하며, 일관성 있고, 완전한지 ( ② )이 필요</li> </ul>	요구사항 확인	<ul style="list-style-type: none"> <li>① : 확인</li> <li>② : 검증</li> </ul>

02. 요구사항 확인

11	요구사항 확인 기법	요구사항 확인	<ul style="list-style-type: none"> <li>* 동료 검사                             <ul style="list-style-type: none"> <li>- 요구사항 명세서 작성자가 요구사항 명세서를 설명하고 <u>이해관계자들이 설명을 들으면</u> 서 결함을 발견하는 형태로 진행</li> </ul> </li> <li>* 워크 스루                             <ul style="list-style-type: none"> <li>- 검토 자료를 회의 전에 배포해 사전검토 후 짧은 시간 동안 회의 진행하는 형태로 리뷰 통해 오류 검출하고 문서화</li> </ul> </li> <li>* 인스펙션                             <ul style="list-style-type: none"> <li>- 소프트웨어 요구, 설계, 원시 코드 등의 저작자 외의 <u>다른 전문가</u>가 또는 팀이 검사하여 오류를 찾아내는 공식적 검토 방법</li> </ul> </li> </ul>
12	요구사항 확인 기법의 종류를 서술하십시오	요구사항 확인	<ol style="list-style-type: none"> <li>1. 요구사항 검토</li> <li>2. 프로토타이핑</li> <li>3. 모델 검증</li> <li>4. 인수 테스트</li> </ol>
13	요구사항 검토에 대해 서술하십시오	요구사항 확인	<ul style="list-style-type: none"> <li>* 여러 검토자들이 에러, 잘못된 가정, 불명확성, 표준과의 차이 등을 검토</li> <li>* 고객 중심 프로젝트에서는 검토자 그룹에 고객 대표자 1명 이상 포함 필요</li> <li>* 시스템 정의서, 시스템 사양서, 소프트웨어 요구사항 명세서를 완성한 시점에서 검토</li> </ul>
14	프로토타이핑에 대해 서술하십시오	요구사항 확인	<ul style="list-style-type: none"> <li>* 새로운 요구사항을 도출하기 위한 수단 또는 소프트웨어 요구사항에 대해 소프트웨어 엔지니어가 해석한 것을 확인하기 위한 수단으로 사용하는 기법이다</li> </ul>
15	모델 검증에 대해 서술하십시오	요구사항 확인	<ul style="list-style-type: none"> <li>* 분석 단계에서 개발된 모델의 품질 검증 필요</li> <li>* 객체 모델의 경우 객체들 사이의 존재하는 의사소통 경로를 검증하기 위한 정적 분석 수행에 유용</li> </ul>
16	인수 테스트에 대해 서술하십시오	요구사항 확인	<ul style="list-style-type: none"> <li>* 요구사항의 중요한 속성은 최종 제품을 기준으로 요구사항을 만족시키는지 확인이 가능해야 함</li> <li>* 각각의 요구사항을 어떻게 확인할 것인지에 대한 계획이 필요</li> </ul>
17	요구사항 관리 프로세스 순서대로 나열	요구사항 관리 프로세스	<ol style="list-style-type: none"> <li>1. 요구사항 협상</li> <li>2. 요구사항 기준선</li> <li>3. 요구사항 변경 관리</li> <li>4. 요구사항 확인 및 검증</li> </ol>
18	요구사항 관리 프로세스 순서	요구사항 관리 프로세스	<ol style="list-style-type: none"> <li>1. 요구사항 협상                             <ul style="list-style-type: none"> <li>- 우선순위 설정</li> <li>- 시뮬레이션</li> </ul> </li> <li>2. 요구사항 기준선                             <ul style="list-style-type: none"> <li>- 공식 회의</li> <li>- 형상 관리</li> <li>- 공식적으로 검토되고 합의된 요구사항 명세서</li> </ul> </li> <li>3. 요구사항 변경 관리                             <ul style="list-style-type: none"> <li>- CCB</li> <li>- 영향도 분석</li> <li>- 요구사항 기준선을 기반으로 모든 변경을 공식적으로 통제하기 위한 기법</li> </ul> </li> <li>4. 요구사항 확인 및 검증                             <ul style="list-style-type: none"> <li>- 확인 및 검증</li> </ul> </li> </ol>

		19	요구사항 확인 프로세스 종류를 서술하시오	요구사항 확인 프로세스	1. 요구사항 목록 확인 2. 요구사항 정의서 작성 여부 확인 3. 비기능적 요구사항 확인 4. 타 시스템 연계 및 인터페이스 요구사항 확인
		20	비기능적 요구사항의 확인 설명	요구사항 확인 프로세스	* 시스템 특성, 품질, 제약사항 등 비기능적 요구사항이 명확하게 도출되었는지 검토 * 성능, 가용성, 사용 용이성, 유지보수 용이성, 안전성, 보안성 등에 대한 요구사항의 문서화 여부 확인
	2. 요구사항의 시스템화 타당성 분석	1	상호 운용성의 개념에 대해 서술하시오	기술적 타당성 검토	다른 목적을 지닌 2개 이상 시스템들이 상호 간 정보 및 서비스를 교환하면서 효과적으로 운용될 수 있는 시스템의 능력이다
		2	기술적 타당성 검토 항목을 서술하시오	기술적 타당성 검토	1. 성능 및 용량 산정의 적정성 2. 시스템 간 상호 운용성 3. IT 시장 성숙도 및 트렌드 부합성 4. 기술적 위험 분석
		3	프로젝트 관리자에 대해 설명하시오	기술적 타당성 검토	요구사항의 기술적 타당성 분석 결과에 대해 이해관계자들의 이견이 있는 경우 중재하여 합의를 도출해 내는 사람
		4	기술적 타당성 분석 프로세스를 순서대로 나열하시오	기술적 타당성 분석	1. 타당성 분석 결과 기록 2. 타당성 분석 결과의 이해관계자 검증 3. 타당성 분석 결과 확인 및 배포 공유
		5	하향식 산정방법(↓) 서술	비용산정 모델	1. 전문가 판단 2. 델파이 기법 - 조정자
		6	상향식 산정방법(↑) 서술	비용산정 모델	1. 코드 라인 수(LoC) - (낙관치 + 4 * 중간치 + 비관치) / 6 2. Man Month 3. COCOMO 모형 - 보형, 단순형/중간형/임베디드형 4. Putnam 모형 - 인력 분포 가정 5. FP (Function Point) 모형
		7	Man Month 구하기	비용산정 모델	Man Month = LoC / 프로그래머 월간 생산성
		8	프로젝트 기간	비용산정 모델	프로젝트 기간 = Man Month / 프로젝트 인력
1. 분석 모델 검증		1	분석 모델 검증 방법의 종류	분석 모델 검증 방법	1. 유스케이스 모델 검증 2. 개념 수준의 분석 클래스 검증 3. 분석 클래스 검증
		2	유스케이스 모델 검증 방법에 대해 서술	분석 모델 검증 방법	시스템 기능에 대한 유스케이스 모형 상세화 수준 및 적정성 검증을 위해서 액터, 유스케이스, 유스케이스 명세서를 점검하는 기법이다
		3	개념 수준의 분석 클래스 검증 방법에 대해 서술	분석 모델 검증	* 시스템의 주요 도메인 개념을 분석 클래스로 도출하여 유스케이스 분석에 활용하므로, 개념 수준의 주요 분석 클래스를 적절히 도출하였는지, 관련 정보가 명확하지를 점검함 * 주요 클래스 도출 여부, 도출된 클래스 이름과 속성의 적절성, 올바른 클래스 간의 관계 여부를 점검
		4	역할 구분 설명	분석 클래스의 스테레오 타입	* 경계 (Boundary) - 시스템과 외부 액터와의 상호작용 담당하는 클래스  * 엔티티 (Entity) - 시스템이 유지해야 하는 정보 관리하는 기능을 전담하는 클래스  * 제어 (Control) - 시스템이 제공하는 기능의 로직 및 제어를 담당하는 클래스
		5	분석 모델 검증 위한 프로세스 설명	분석 모델 검증	1. 검토의견 컬럼 추가 2. 검토의견 작성 3. 검토의견 정제

03. 분석 모델 확인하기		6	분석 모델 검증 위한 프로세스 설명	분석 모델 검증	<ul style="list-style-type: none"> <li>1. 검토의견 컬럼 추가 <ul style="list-style-type: none"> <li>- 분석 모델까지 요구사항 추적표를 작성하고 검토의견 컬럼 추가</li> </ul> </li> <li>2. 검토의견 작성 <ul style="list-style-type: none"> <li>- 요구사항 목록 참조하여 요구사항 ID와 요구사항명 입력</li> <li>- 유스케이스 모델에 대한 검토의견 작성</li> <li>- 개념 수준의 분석 클래스 모델에 대한 검토의견 작성</li> <li>- 분석 클래스 모델에 대한 검토의견 작성</li> </ul> </li> <li>3. 검토의견 정제 <ul style="list-style-type: none"> <li>- 요구사항 추적표에서 요구사항에 대한 검토의견 정제</li> <li>- 누락된 유스케이스 모델/개념 수준 분석 클래스/분석 클래스가 존재하는 경우, 검토 의견 추가</li> </ul> </li> </ul>
	2. 분석 모델의 시스템화 타당성 분석	1	분석 모델의 기술적 타당성 검토 항목 서술	분석 모델의 기술적 타당성 검토	<ul style="list-style-type: none"> <li>1. 성능 및 용량 산정의 적정성</li> <li>2. 시스템 간의 상호 운용성</li> <li>3. IT 시장 성숙도 및 트렌드 부합성</li> <li>4. 기술적 위험 분석</li> </ul>
		2	성능 및 용량 산정의 적정성에 대해 설명하십시오	분석 모델의 기술적 타당성 검토	<ul style="list-style-type: none"> <li>* 요구사항을 만족시키기 위한 분석 모델에 따라 시스템을 구현할 때 요구되는 시스템의 자원 식별</li> <li>* 분석 클래스에서 불필요하고 지나치게 많은 속성들을 포함시키게 되면, 객체 생성 시 시스템의 메모리 자원이 많이 요구되며, 전체 시스템의 성능 저하 발생</li> </ul>
		3	기술적 위험 분석에 대해 설명하십시오	분석 모델의 기술적 타당성 검토	<ul style="list-style-type: none"> <li>* 분석 모델이 시스템의 기술 구조, 프레임워크, 사용되는 하드웨어 및 소프트웨어와 부합되는지 확인하는 기법</li> <li>* 분석 모델이 검증되지 않은 기술의 사용을 가정으로 하고 있어 추가적인 비용 발생 가능성이 있는지 확인하는 기법</li> <li>* 분석 모델을 구현하기 위하여 특정 업체 기술, 특허, 라이선스에 의존해야 하는지 확인하는 기법</li> </ul>
		4	분석모델의 시스템화 타당성 분석 프로세스 순서대로 나열	시스템화 타당성 분석 프로세스	<ul style="list-style-type: none"> <li>1. 타당성 검토의견 컬럼 추가</li> <li>2. 타당성 검토의견 작성</li> <li>3. 타당성 분석 결과 검증</li> <li>4. 타당성 분석 결과 확인 및 배포/공유</li> </ul>
		5	분석모델의 시스템화 타당성 분석 프로세스 순서대로 나열	시스템화 타당성 분석 프로세스	<ul style="list-style-type: none"> <li>1. 타당성 검토의견 컬럼 추가 <ul style="list-style-type: none"> <li>- 분석 모델까지 요구사항 추적표를 작성하고, 타당성 검토의견 컬럼을 추가</li> </ul> </li> <li>2. 타당성 검토의견 작성 <ul style="list-style-type: none"> <li>- 작성된 요구사항 추적표에 타당성 검토의견 작성</li> <li>- 타당성 검토의견 제외한 나머지 속성들은 분석 모델 검증 수행 내용의 작성 절차와 동일</li> </ul> </li> <li>3. 타당성 분석 결과 검증 <ul style="list-style-type: none"> <li>- 타당성 분석 결과를 관련 이해관계자에게 배포하여 사전 검토를 요청</li> <li>- 관련 이해관계자가 모여 분석 모델 타당성 분석 결과를 검증</li> </ul> </li> <li>4. 타당성 분석 결과 확인 및 배포/공유 <ul style="list-style-type: none"> <li>- 이해관계자 검증을 거친 타당성 분석 결과를 의사 결정자 확인</li> <li>- 확정된 타당성 분석 결과를 이해관계자에게 배포하여 공유</li> </ul> </li> </ul>