

직무명 : 응용SW엔지니어링

과목명 : 웹 기반 애플리케이션 구현

1. 직무 개요

1) 직무 정의

응용소프트웨어 엔지니어링은 컴퓨터 프로그래밍 언어로 각 업무에 맞는 소프트웨어의 기능에 관한 설계, 구현 및 테스트를 수행하고, 사용자에게 배포하며, 버전관리를 통해 제품의 성능을 향상시키고, 서비스를 개선하는 일이다.

2) 능력단위

순 번	능 력 단 위
1	요구사항 확인
2	데이터 입출력 구현
3	통합 구현
4	정보시스템 이행
5	제품소프트웨어 패키징
6	서버프로그램 구현
7	인터페이스 구현
8	애플리케이션 배포
9	프로그래밍 언어 활용
10	응용 SW 기초 기술 활용
11	애플리케이션 리팩토링
12	인터페이스 설계
13	애플리케이션 요구사항 분석
14	기능 모델링
15	애플리케이션 설계
16	정적모델 설계
17	동적모델 설계
18	화면 설계
19	화면 구현
20	애플리케이션 테스트 관리
21	애플리케이션 테스트 수행
22	소프트웨어공학 활용
23	소프트웨어개발 방법론 활용

3) 능력단위별 능력단위요소

분 류 번 호	능 력 단 위 명	수 준	능 력 단 위 요 소
2001020201_16v3	요구사항 확인	5	현행 시스템 분석하기
			요구사항 확인하기
			분석모델 확인하기
2001020205_16v4	데이터 입출력 구현	5	논리 데이터저장소 확인하기
			물리 데이터저장소 설계하기
			데이터 조작 프로시저 작성하기
			데이터 조작 프로시저 최적화하기
2001020206_16v4	통합 구현	5	연계 데이터 구성하기
			연계 매카니즘 구성하기
			내외부 연계 모듈 구현하기
2001020208_16v3	정보시스템 이행	5	정보시스템 사용자 교육하기
			정보시스템 이행하기
			정보시스템 안정화하기
2001020209_16v4	제품소프트웨어 패키징	5	제품소프트웨어 패키징하기
			제품소프트웨어 매뉴얼 작성하기
			제품소프트웨어 버전관리하기
2001020211_16v4	서버프로그램 구현	5	개발환경 구축하기
			공통 모듈 구현하기
			서버 프로그램 구현하기
			배치 프로그램 구현하기
2001020212_16v4	인터페이스 구현	5	인터페이스 설계서 확인하기
			인터페이스 기능 구현하기
			인터페이스 구현 검증하기
2001020214_16v4	애플리케이션 배포	3	애플리케이션 배포 환경 구성하기
			애플리케이션 소스 검증하기
			애플리케이션 빌드하기
			애플리케이션 배포하기
2001020215_15v3	프로그래밍 언어 활용	3	기본문법 활용하기
			언어특성 활용하기
			라이브러리 활용하기
2001020216_15v3	응용 SW 기초 기술 활용	3	운영체제 기초 활용하기
			데이터베이스 기초 활용하기
			네트워크 기초 활용하기
			기본 개발환경 구축하기

분 류 번 호	능 력 단 위 명	수 준	능 력 단 위 요 소
2001020217_16v4	애플리케이션 리팩토링	6	리팩토링 기준 수립하기
			소스 코드 분석하기
			리팩토링 적용하기
2001020218_16v4	인터페이스 설계	6	인터페이스 요구사항 확인하기
			인터페이스 대상 식별하기
			인터페이스 상세 설계하기
2001020219_16v4	애플리케이션 요구사항 분석	7	요구사항 도출하기
			요구사항 분석하기
			요구사항 명세화하기
			요구사항 검증하기
2001020220_16v1	기능 모델링	6	기능 모델 작성하기
			기능 모델 검증하기
2001020221_16v4	애플리케이션 설계	6	공통 모듈 설계하기
			타 시스템 연동설계하기
2001020222_16v4	정적모델 설계	6	정적 분석모델 검증하기
			정적모델 상세화하기
2001020223_16v4	동적모델 설계	6	동적 분석모델 검증하기
			동적모델 상세화하기
2001020224_16v4	화면 설계	5	UI 요구사항 확인하기
			UI 설계하기
2001020225_16v4	화면 구현	3	UI 설계 확인하기
			UI 구현하기
2001020226_16v4	애플리케이션 테스트 관리	5	애플리케이션테스트케이스 설계하기
			애플리케이션 통합 테스트하기
			애플리케이션 성능 개선하기
2001020227_16v4	애플리케이션 테스트 수행	3	애플리케이션 테스트 수행하기
			애플리케이션 결함 조치하기
2001020228_16v4	소프트웨어공학 활용	6	CASE 도구 활용하기
			품질요구사항 확인하기
2001020229_16v4	소프트웨어개발 방법론 활용	6	소프트웨어개발 방법론 선정하기
			소프트웨어개발방법론 테일러링하기

2. 훈련이수체계(수준별 이수 과정/과목)

8	임원	
7	부장 차장	애플리케이션 요구사항 분석
6	과장 대리	애플리케이션 리팩토링 인터페이스 설계 기능모델링 애플리케이션 설계 장치모델 설계 동적모델 설계 소프트웨어공학 활용 소프트웨어 개발방법론 활용
5	사원	요구사항확인 데이터 입출력 구현 통합 구현 정보시스템 이행 제품소프트웨어 패키징 서버프로그램 구현 인터페이스 구현 화면설계 애플리케이션 테스트관리
4	사원	
3	사원	프로그래밍 언어 활용 응용SW 기초기술 활용 애플리케이션 배포 화면 구현 애플리케이션 테스트 수행
작업준 작업형		응용SW 엔지니어링

분류번호 : 2001020205_16v4

능력단위 명칭 : 데이터 입출력 구현

능력단위 정의 : 데이터 입출력 구현이란 응용소프트웨어가 다루어야 하는 데이터 및 이들 간의 연관성, 제약조건을 식별하여 논리적으로 조직화 하고, 소프트웨어 아키텍처에 기술된 데이터저장소에 조직화된 단위의 데이터가 저장될 최적화된 물리적 공간을 구성하고 데이터 조작언어를 이용하여 구현하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020205_16v4.1 논리 데이터저장소 확인하기	1.1 업무 분석가, 데이터베이스 엔지니어가 작성한 논리 데이터저장소 설계 내역에서 정의된 데이터의 유형을 확인하고 식별할 수 있다. 1.2 논리 데이터저장소 설계 내역에서 데이터의 논리적 단위와 데이터 간의 관계를 확인할 수 있다. 1.3 논리 데이터저장소 설계 내역에서 데이터 또는 데이터간의 제약조건과 이들 간의 관계를 식별할 수 있다.
	【지 식】 <ul style="list-style-type: none">○ 구현에 요구되는 아키텍처의 이해○ 설계 산출물의 이해○ 데이터베이스에 대한 이해○ 자료구조○ 수학, 통계 지식○ E-R Modeling○ SQL(Structure Query Language)
	【기 술】 <ul style="list-style-type: none">○ E-R 모델링 도구 활용 능력○ UML 모델링 도구 활용 능력○ 모델링 검증 도구 활용 능력○ DBMS 사용 기술○ DBMS 관리 도구 활용 능력○ ORM 프레임워크 활용 능력
2001020205_16v4.2	【태 도】 <ul style="list-style-type: none">○ 정확성과 완전성을 기하고자 하는 의지○ 산출물 완성도를 위한 적극적인 태도○ 기술 및 제품에 대한 정보수집과 학습에 대한 적극성○ 자신이 수행한 작업에 대한 평가의 객관성○ 데이터베이스 기술 동향에 대한 이해와 관련 제품들에 대해 파악하기 위한 노력
	2.1 논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을

능 력 단 위 요 소	수 행 준 거
물리 데이터저장소 설계하기	반영한 물리 데이터저장소 설계를 수행할 수 있다. 2.2 논리 데이터저장소 설계를 바탕으로 목표 시스템의 데이터 특성을 반영하여 최적화된 물리 데이터저장소를 설계할 수 있다. 2.3 물리 데이터저장소 설계에 따라 데이터저장소에 실제 데이터가 저장될 물리적 공간을 구성할 수 있다.
	【지 식】 <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처의 이해 ○ 설계 산출물의 이해 ○ 자료구조 ○ 데이터베이스에 대한 이해 ○ 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등) ○ E-R Modeling ○ SQL(Structure Query Language)
	【기 술】 <ul style="list-style-type: none"> ○ E-R 모델링 도구 활용능력 ○ UML 모델링 도구 활용 능력 ○ 모델링 검증 도구 활용 능력 ○ RDBMS를 통한 구현 능력 ○ 트랜잭션 인터페이스 설계 능력 ○ ORM 프레임워크 활용 능력
2001020205_16v4.3 데이터 조작 프로시저 작성하기	【태 도】 <ul style="list-style-type: none"> ○ 정확성과 완전성을 기하고자 하는 의지 ○ 산출물 완성도를 위한 적극적인 태도 ○ 기술 및 제품에 대한 정보수집과 학습에 대한 적극성 ○ 자신이 수행한 작업에 대한 평가의 객관성 ○ 데이터베이스 기술 동향에 대한 이해와 관련 제품들에 대해 파악하기 위한 노력
	3.1 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소에 연결을 수행하는 프로시저를 작성할 수 있다. 3.2 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소로부터 데이터를 읽어 오는 프로시저를 작성할 수 있다. 3.3 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 변경 내용 또는 신규 입력된 데이터를 데이터 저장소에 저장하는 프로시저를 작성할 수 있다. 3.4 구현된 데이터 조작 프로시저를 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.
	【지 식】 <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처의 이해 ○ 설계 산출물의 이해 ○ 프로그래밍언어(C, C++, Java-JSP/Servlet 등)와 도구(IDE 등)의 활용방법에 대한 이해 ○ 데이터베이스에 대한 이해

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등) ○ SQL(Structure Query Language) ○ 소프트웨어 테스트에 대한 이해
	【기 술】 <ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ ORM 프레임워크 활용 능력 ○ 단위테스트 도구 활용기법 ○ 쿼리(Query) 성능 측정 도구 활용 능력
	【태 도】 <ul style="list-style-type: none"> ○ 기술 및 제품에 대한 정보수집과 학습에 대한 적극성 ○ 정확성과 완전성을 기하고자 하는 의지 ○ 기술 및 제품에 대한 정보수집과 학습에 대한 적극성 ○ 자신이 수행한 작업에 대한 평가의 객관성
2001020205_16v4.4 데이터 조작 프로시저 최적화하기	4.1 프로그래밍 언어와 도구에 대한 이해를 바탕으로 응용소프트웨어 설계, 물리 데이터저장소 설계와 운영 환경을 고려하여 데이터 조작 프로시저의 성능을 예측할 수 있다. 4.2 업무 분석가에 의해 정의된 요구사항을 기준으로, 성능측정 도구를 활용하여 데이터 조작 프로시저의 성능을 측정할 수 있다. 4.3 실 데이터를 기반으로 테스트를 수행하여 데이터 조작 프로시저의 성능에 영향을 주는 병목을 파악할 수 있다. 4.4 테스트 결과와 정의된 요구사항을 기준으로 데이터조작 프로시저의 성능에 따른 이슈 발생 시 이에 대해 해결할 수 있다.
	【지 식】 <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처의 이해 ○ 프로그래밍언어(C, C++, Java-JSP/Servlet 등)와 도구(IDE 등)의 활용방법의 이해 ○ 성능 평가 기준에 대한 이해 ○ 데이터베이스에 대한 이해 ○ 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등) ○ SQL(Structure Query Language) ○ 소프트웨어 테스트에 대한 이해
	【기 술】 <ul style="list-style-type: none"> ○ 소스코드 표준 문서화 능력 ○ 개발에 필요한 프로그래밍 언어 표준과 코드 검토 기법 ○ 성능 측정 도구 활용 능력 ○ 표준 단어, 용어, 도메인, 코드에 대한 정의 능력 ○ 소스코드 인스펙션 능력 ○ 쿼리(Query) 성능 측정 도구 활용 능력

능 력 단 위 요 소	수 행 준 거
	<p>【태 도】</p> <ul style="list-style-type: none"> ○ 개발 수행에 요구되는 표준을 준수하려는 태도 ○ 제품의 품질 및 고객 우선주의 태도 ○ 정확성과 완전성을 기하고자 하는 의지 ○ 자신이 수행한 작업에 대한 평가의 객관성 ○ 객관적이고 논리적인 평가를 수행할 수 있는 태도

분류번호 : 2001020211_16v4

능력단위 명칭 : 서버프로그램 구현

능력단위 정의 : 서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020211_16v4.1 개발환경 구축하기	<p>1.1 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.</p> <p>1.2 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다.</p> <p>1.3 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 개발에 필요한 하드웨어 및 소프트웨어 사양○ 개발환경에 요구되는 아키텍처○ 서버 개발 프레임워크 개념○ 설계 산출물 해석 방법○ 데이터베이스 개념○ 네트워크 개념○ 보안 관련 개념 <p>【기 술】</p> <ul style="list-style-type: none">○ 컴파일러 사용 능력○ IDE 도구 활용 능력○ 네트워크 활용 기술○ 협업도구 활용 능력○ 형상관리 도구 활용 능력○ 프레임워크 활용 기술○ 프로그래밍언어 활용 기술○ 보안도구 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none">○ 개발 수행에 요구되는 표준을 준수하려는 태도○ 성공적인 개발을 위한 의지○ 산출물 완성도를 위한 적극적인 태도○ 개발 팀원 간의 원활한 협업을 추구하는 태도○ 주어진 과제를 완수하는 책임감
2001020211_16v4.2 공통 모듈 구현하기	<p>2.1 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>2.2 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다.</p> <p>2.3 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 개발환경에 요구되는 아키텍처 ○ 설계 산출물 해석 방법 ○ 프로그램 언어 활용방법 ○ 데이터베이스 개념 ○ 네트워크 개념 ○ 분산 처리 개념 ○ 소프트웨어 테스트 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등) ○ 재사용 기법 ○ 모듈화 기법 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 개발 수행에 요구되는 표준을 준수하려는 태도 ○ 성공적인 개발을 위한 의지 ○ 산출물 완성도를 위한 적극적인 태도 ○ 개발 팀원 간의 원활한 협업을 추구하는 태도 ○ 주어진 과제를 완수하는 책임감
<p>2001020211_16v4.3</p> <p>서버 프로그램 구현하기</p>	<p>3.1 업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다.</p> <p>3.2 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다.</p> <p>3.3 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다.</p> <p>3.4 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 설계 산출물 해석방법 ○ 프로그램 언어 활용방법 ○ 단위 업무 ○ 단위테스트 기술동향 및 도구

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 소프트웨어 테스트 ○ 결함 관리 【기 술】 <ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ IDE 도구 활용 능력 ○ 프로그램 디버깅 기술 ○ 프로그램 코드 검토 기술 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등) 【태 도】 <ul style="list-style-type: none"> ○ 개발 수행에 요구되는 표준을 준수하려는 태도 ○ 제품의 품질 및 고객 우선주의 태도 ○ 적극적이고 세밀히 분석하는 자세 ○ 기술 및 도구에 대해 적극적으로 배우고자 하는 태도 ○ 작업환경 개선 의지
2001020211_16v4.4 배치 프로그램 구현하기	4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다. 4.2 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다. 4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.
	【지 식】 <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 설계 산출물 해석방법 ○ 프로그램 언어 활용방법 ○ 데이터베이스 개념 ○ 네트워크 개념 ○ 분산 프로그래밍 개념 ○ 소프트웨어 테스트 【기 술】 <ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 인터페이스 도구/기법 (API, EAI, WebService, 등) 【태 도】 <ul style="list-style-type: none"> ○ 개발 수행에 요구되는 표준을 준수하려는 태도 ○ 성공적인 개발을 위한 의지 ○ 산출물 완성도를 위한 적극적인 태도 ○ 개발 팀원 간의 원활한 협업을 추구하는 태도 ○ 주어진 과제를 완수하는 책임감 ○ 적극적인 업무 분석 태도

분류번호 : 2001020214_16v4

능력단위 명칭 : 애플리케이션 배포

능력단위 정의 : 애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020214_16v4.1 애플리케이션 배포 환경 구성하기	<p>1.1 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다.</p> <p>1.2 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다.</p> <p>1.3 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다.</p> <p>1.4 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○애플리케이션 빌드(Build) 절차○애플리케이션 배포 환경 구성 지식○형상관리 도구 설치 및 사용법○빌드 도구 설치 및 사용법○배포 도구 설치 및 사용법 <p>【기 술】</p> <ul style="list-style-type: none">○ 형상관리 도구 명령어○ 빌드 도구 명령어○ 빌드 스크립트 작성 기술○ 배포 도구 명령어○ 배포 스크립트 작성 기술○ 환경 설정 기법 <p>【태 도】</p> <ul style="list-style-type: none">○ 다양한 도구와 연동을 통한 최적의 배포 환경을 구성하고자 하는 태도○ 애플리케이션의 특성을 고려한 배포 환경을 구성하고자 하는 자세○ 자동화 도구를 효율적으로 다룰 수 있는 자세
2001020214_16v4.2 애플리케이션 소스 검증하기	<p>2.1 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다.</p> <p>2.2 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다.</p> <p>2.3 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.</p> <p>【지 식】</p>

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 형상관리 도구 사용법 ○ 형상관리 지침 ○ 소스코드 검증 기준 ○ 소스코드 검증 기법 ○ 소스코드 검증 도구 ○ 소스코드 정적 검증 분석 기법 ○ 소스코드 동적 검증 분석 기법 【기 술】 <ul style="list-style-type: none"> ○ 소스코드 확보 기술 ○ 소스코드 검증 기술 ○ 소스코드 검증 도구 사용 기술 ○ 소스코드 검증 도구 결과 분석 기술 【태 도】 <ul style="list-style-type: none"> ○ 애플리케이션의 품질 향상을 위해 다양한 관점에서 소스코드를 검증하는 자세 ○ 자동화 도구를 활용하여 소스코드를 검증하는 노력 ○ 자동화 도구 외 수작업으로 검토 및 리뷰를 통해 검증을 실시하는 자세 ○ 소스코드 검증 결과 문제 발생시 해결하고자 하는 적극적인 자세 ○ 다양한 분석도구를 다루면서 효율적으로 활용하고자 하는 자세
2001020214_16v4.3 애플리케이션 빌드하기	<p>3.1 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다.</p> <p>3.2 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다.</p> <p>3.3 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다.</p> <p>3.4 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지 여부를 확인할 수 있다.</p> <p>3.5 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명할 수 있다.</p>
	【지 식】 <ul style="list-style-type: none"> ○ 애플리케이션 빌드 시스템 가이드 ○ 애플리케이션 빌드 절차 ○ 애플리케이션 빌드 스크립트 문법 ○ 빌드 결과 분석 지식 【기 술】 <ul style="list-style-type: none"> ○ 애플리케이션 빌드 스크립트 작성 기법 ○ 애플리케이션 빌드 도구 적용 기술 ○ 빌드 실행 기술 ○ 빌드 검증 기술 【태 도】

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ◦ 검증이 완료된 소스코드는 빌드 시스템으로 정확하게 이관하는 태도 ◦ 빌드 절차를 준수하여 성공적으로 빌드를 수행하고자하는 자세 ◦ 빌드 결과를 이해하고 문제 발생 시 즉시 조치하는 태도 ◦ 빌드 진행 상태를 모니터링 하면서 결과를 확인하는 태도 ◦ 빌드 실패 시 원인 분석을 실시하여 개발자에서 설명하는 자세
2001020214_16v4.4 애플리케이션 배포하기	4.1 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다. 4.2 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다. 4.3 애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다. 4.4 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.
	【지 식】 <ul style="list-style-type: none"> ◦ 애플리케이션 운영 환경 정보 ◦ 애플리케이션 운영 플랫폼 정보 ◦ 애플리케이션 운영 환경 배포 절차 ◦ 애플리케이션 원복 절차 【기 술】 <ul style="list-style-type: none"> ◦ 애플리케이션 운영 환경 배포 기술 ◦ 애플리케이션 운영 환경 적용 도구 활용 기술 ◦ 운영 배포 결과 확인 기법 ◦ 애플리케이션 원복 절차 ◦ 애플리케이션 원복 실행 도구 사용법 【태 도】 <ul style="list-style-type: none"> ◦ 애플리케이션 배포 이전에 정확한 환경 정보로 구성했는지 여부를 확인하는 태도 ◦ 애플리케이션 배포 이전에 빌드 후 패키징 결과물의 이상 여부를 검증하는 자세 ◦ 애플리케이션 배포 후 정상 작동 여부를 확인하는 자세 ◦ 배포 시 문제 발생 시 즉시 원인 파악을 하고자 하는 자세 ◦ 문제 발생 애플리케이션에 대한 원상 복구 여부를 판단하는 자세 ◦ 애플리케이션 원복 결정 시 즉시 절차에 따라 실행하는 태도

분류번호 : 2001020227_16v4

능력단위 명칭 : 애플리케이션 테스트 수행

능력단위 정의 : 애플리케이션 테스트 수행이란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트 케이스에 따라 테스트를 수행하고 결함을 조치하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020227_16v4.1 애플리케이션 테스트 수행하기	<p>1.1 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.</p> <p>1.2 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다.</p> <p>1.3 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 산업 분야별 업무특성○ 소프트웨어 아키텍처 국제 표준(IEEE 1471)○ 국제 표준 제품 품질특성(ISO 25000)○ 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)○ 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법○ 결함 관리 <p>【기 술】</p> <ul style="list-style-type: none">○ 컴파일러 사용법○ IDE 도구 활용 능력○ 프로그램 디버깅 기법○ 프로그램 코드 검토 기법○ 형상관리 도구 활용 기법○ 테스트 자동화 도구 활용기법○ APM(Application Performance Management) 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 태도○ 제품의 품질 및 고객 우선주의 태도○ 기술 및 제품 동향에 관한 각종 정보 수집에 대한 적극성○ 제품에 대한 책임감 및 불량에 대한 재발 방지를 위해 노력하는 태도○ 적극적이고 세밀히 분석하는 자세
2001020227_16v4.2 애플리케이션 결함 조치하기	<p>2.1 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.</p> <p>2.2 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>2.3 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 산업 분야별 업무특성 ○ 소프트웨어 아키텍처 국제 표준(IEEE 1471) ○ 국제 표준 제품 품질특성(ISO 25000) ○ 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK) ○ 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법 ○ 결함 관리 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 컴파일러 사용법 ○ IDE 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 프로그램 코드 검토 기법 ○ 형상관리 도구 활용 기법 ○ 테스트 자동화 도구 활용기법 ○ APM(Application Performance Management) 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 태도 ○ 제품의 품질 및 고객 우선주의 태도 ○ 기술 및 제품 동향에 관한 각종 정보 수집에 대한 적극성 ○ 제품에 대한 책임감 및 불량에 대한 재발 방지를 위해 노력하는 태도 ○ 적극적이고 세밀히 분석하는 자세