

CHAPTER II

환경분석

1 노동시장분석

1 산업현장 직무능력수준

세분류 직능수준	01.SW아키텍처	02.응용SW엔지니어링	03.시스템엔지니어링	04.DB엔지니어링
VII(직무경험: 15~20년)	총괄 아키텍트			
VI(직무경험: 10~14년)	책임 SW아키텍트	응용SW 통합개발자	특급 임베디드SW 개발자	DB엔지니어링 책임자
V(직무경험: 6~9년)	선임 SW아키텍트	응용SW 분석개발자	고급 임베디드SW 개발자	DB엔지니어링 담당자
IV(직무경험: 3~5년)	SW아키텍처 설계 담당자	응용SW 실무개발자	중급 임베디드SW 개발자	DB엔지니어링 실무자
III(직무경험: 1~2년)	SW아키텍처 구현 담당자	패키징 실무자	초급 임베디드SW 개발자	DB엔지니어링 기능자
II(직무경험: 0~1년)		UI개발자		

세분류 직능수준	05.NW엔지니어링	06.보안엔지니어링	07.UI/UX엔지니어링
VII(직무경험: 15~20년)		보안 총괄 관리자	
VI(직무경험: 10~14년)	NW 프로젝트 관리자 / NW 시스템 관리자	보안 책임자	UI/UX 총괄 관리자
V(직무경험: 6~9년)	NW SW 설계자	보안 관리자	리서치 담당자 / UI/UX 기획 담당자 / UI 설계 담당자 /
IV(직무경험: 3~5년)	NW SW 품질 담당자	보안 담당자	UI 디자인 담당자
III(직무경험: 1~2년)	NW 프로그래머	보안 실무자	UI 테스트 담당자 / UI 구현 담당자
II(직무경험: 0~1년)			

2 사업체 및 종사자 수

소분류	세분류	관련업종	사업체수	종사자수
2. 정보기술개발	기재생략	소프트웨어 개발 및 공급업	9,657	152,170
		게임 소프트웨어 개발 및 공급업	1,413	32,992
		온라인·모바일 게임 소프트웨어 개발 및 공급업	1,145	30,175
		기타 게임 소프트웨어 개발 및 공급업	268	2,817
		시스템·응용 소프트웨어 개발 및 공급업	8,244	119,178
		시스템 소프트웨어 개발 및 공급업	4,489	73,432
		응용소프트웨어 개발 및 공급업	3,755	45,746
		컴퓨터 프로그래밍, 시스템 통합 및 관리업	5,779	83,265
		컴퓨터 프로그래밍 서비스업	3,761	28,310
		컴퓨터시스템 통합 자문, 구축 및 관리업	1,562	50,247
		컴퓨터 시스템 통합 자문 및 구축 서비스업	1,247	45,809
		컴퓨터시설 관리업	315	4,438
		기타 정보기술 및 컴퓨터운영 관련 서비스업	456	4,708
		정보서비스업	2,378	39,533
		자료처리, 호스팅, 포털 및 기타 인터넷 정보매개서비스업	1,389	24,775
		자료처리, 호스팅 및 관련 서비스업	329	6,643
		자료 처리업	201	3,847
		호스팅 및 관련 서비스업	128	2,796
		포털 및 기타 인터넷 정보매개 서비스업	1,060	18,132
		기타 정보 서비스업	989	14,758
		뉴스 제공업	180	1,485
		그 외 기타 정보 서비스업	809	13,273
		데이터베이스 및 온라인정보 제공업	791	12,868
		그 외 기타 정보 서비스업	18	405
합	계		50,363	811,802

※ 상기 사업체 수와 종사자 수는 통계청 웹사이트(<http://kosis.kr>) 의 “2013년 전국사업체 조사 (시도·산업·종사자규모별 사업체수, 종사자수) ” 및 “ 2011년 정보통신기술산업 통계 주요지표 ” 를 근거로 함.

③ 인력배출 현황

중분류	소분류	학 과	교 육 훈 련 기 관	'11년(명)		'12년(명)		'13년(명)	
				입 학	졸 업	입 학	졸 업	입 학	졸 업
1. 정보기술	02. 정보기술 개발	전산/컴퓨터공학	대학원	1,722	1,591	1,765	1,546	1,572	1,557
			대학	9,789	8,676	10,218	9,118	86,349	8,626
			전문대학	2,341	1,641	2,180	1,781	1,716	1,289
		응용소프트웨어공학	대학원	354	271	289	288	407	319
			대학	1,317	1,076	1,862	967	13,164	1,014
			전문대학	2,207	1,528	2,268	1,478	2,025	1,548
		정보/통신공학	대학원	2,010	1,577	1,946	1,613	1,939	1,579
			대학	10,917	7,840	10,613	7,103	75,051	7,292
			전문대학	12,134	9,607	11,441	8,918	11,013	8,339
		산업공학	대학원	1,100	1,010	1,105	924	1,099	879
			대학	3,287	3,072	3,744	3,056	3,833	2,837
			전문대학	53	4	27	13	37	48
		합 계		47,231	37,893	47,458	36,805	198,205	35,327

※ 출처 : 교육통계연보 (2011~2013년), 한국교육개발원

※ 대학원 입학자는 석사과정 입학 학생수 기준, 졸업자는 석사학위 취득자 수 기준, 전문대학 입학자는 전문학사 입학자 기준임

※ 정보기술 개발 소분류의 입학자 및 졸업자 수는 점진적 감소 추세 경향을 보이고 있음

4 직업정보

세분류		기재 생략											
직업명		정보통신 관련 관리자	컴퓨터 하드웨어 기술자 및 연구원	통신공학 기술자 및 연구원	컴퓨터 시스템 설계 및 분석가	네트워크 시스템 개발자	컴퓨터 보안 전문가	시스템 소프트웨어 개발자	응용 소프트웨어 개발자	웹 개발자	웹 및 멀티미디어 기획자	데이터베이스 개발자	정보 시스템 운영자
종사자 수		4.2천명	5.9천명	36.7천명	11.5천명	6.9천명	4.2천명	72.2천명	91.7천명	10.7천명	4.1천명	5.2천명	53.4천명
종사 현황	연령	평균: 49.5세	평균: 34.5세	평균: 34.5세	평균: 37.3세	평균: 34.7세	평균: 31.8세	평균: 34.9세	평균: 34.7세	평균: 33세	평균: 32세	평균: 35세	평균: 34.7세
	임금	평균: 455.2만원	평균: 338.9만원	평균: 368.5만원	평균: 386.1만원	평균: 311.5만원	평균: 245.6만원	평균: 328.4만원	평균: 301.8만원	평균: 249.1만원	평균: 314.2만원	평균: 324.7만원	평균: 297.1만원
	학력	평균: 14.5년	평균: 16.2년	평균: 16.4년	평균: 16.2년	평균: 15.8년	평균: 15.5년	평균: 15.8년	평균: 15.4년	평균: 15년	평균: 15.4년	평균: 16.1년	평균: 15.5년
	성비	남	92.4%	97.1%	92.9%	80.6%	94.1%	93.0%	84.9%	88.1%	76.4%	75.7%	74.5%
		여	7.6%	2.9%	7.1%	19.4%	5.9%	7.0%	15.1%	11.9%	23.6%	24.3%	25.5%
관련자격	근속년수(평균)	16년	6.7년	6.5년	5.1년	7.2년	4.1년	5.5년	4.4년	3.5년	4.1년	6.7년	6년
		정보통신기술사(국가기술)	전자계산기기능사, 기사, 기술사(국가기술), 전자산업기사(국가기술), 정보통신산업기사, 기사, 기술사(국가기술)	정보통신기술사, 전자응용기술사, 정보통신(산업)기사 등	전자계산기기능사, 기사, 기술사(국가기술), 전자계산조직응용기사, 기술사(국가기술), 정보관리기술사(국가기술)	전자계산조직응용기사/기술사(국가기술), 정보관리기술사(국가기술), 정보처리기사(국가기술), 정보처리기능사, 산업기사(국가기술), 정보관리기술사(국가기술)	CISA(국제공인정보시스템 감사사)(외국), CISSP(국제공인정보시스템 보안전문가)(외국), SIS(정보보호전문가)(국가공인 민간)	전자계산조직응용기사/기술사(국가기술), 정보처리기능사, 산업기사, 기사(국가기술)	전자계산조직응용기사/기술사(국가기술), 정보처리기능사, 산업기사, 기사(국가기술)	OCP(외국), SCJP(외국), 정보처리기능사, 산업기사, 기사(국가기술)	OCP(외국), SCJP(외국), 정보처리기능사, 산업기사, 기사(국가기술)	MCDDBA(외국), OCP(외국), OCP-D BA(외국)	정보처리기술사, 정보처리기사, 산업기사, 전자계산조직응용기술사, 기사, MCSE, OCP, CCIE, CSA

※ 워크넷 (<http://www.work.go.kr>)의 직업정보를 근거로 함.

2 교육훈련 현황 분석

1 교육훈련기관 현황

중분류	소분류	학 과	교육훈련기관		
			구 분	계	교육훈련기관
01. 정보기술	02. 정보기술개발	전산/컴퓨터	대학	200	포항공과대학교, 서강대학교, 가톨릭대학교, 가천대학교, 연세대학교 등
			전문대학	73	인하공업전문대학교, 강동대학교, 선린대학, 경기과학기술대학교 등
			폴리텍대학	7	한국폴리텍 I 대학, 한국폴리텍 II 대학, 한국폴리텍 III 대학 외
			특성화고	59	김포제일고등학교, 동일공업고등학교, 수원전산여자고등학교, 전주공업고등학교, 외
			직업전문학교	4	울산전자직업전문학교, 전자직업전문학교, 서울시립상계직업전문학교 외
		응용소프트웨어공학	대학	54	건국대학교, 경북대학교, 홍익대, 금오공과대학교 등
			전문대학	53	동양미래대학, 계명문화대학, 강원도립대학, 동서울대학 등
			폴리텍대학	1	한국폴리텍대학
			특성화고	16	선린인터넷고, 대진정보통신고 등
		정보통신	대학	123	건국대학교, 고려대학교, 성균관대학교, 성신여자대학교, 포항공과대학교 외
			전문대학	91	강동대학교, 대림대학교, 부산과학기술대학교, 울산과학대학교, 한국정보통신기능대학교 외
			폴리텍대학	8	한국폴리텍 I 대학, 한국폴리텍 II 대학, 한국폴리텍 III 대학 외
			특성화고	305	제천디지털전자고등학교, 부산컴퓨터과학고등학교, 세청도전자고등학교, 봉양정보고등학교 외
			직업전문학교	4	울산전자직업전문학교, 전자직업전문학교, 서울시립상계직업전문학교 외
		산업공학	대학	64	강원대학교, 공주대학교, 부산대학교, 서울대학교, 전남대학교, 건국대학교, 동아대학교 외
			전문대학	5	거제대학교, 구미대학교, 신안산대학교, 경기과학기술대학교, 오산대학교

※ 워크넷 학과정보 / 교육통계서비스('12년)

※ 대학 정보 : http://univ.kcue.or.kr/univinfo/univsetdept_list.do

※ 전문대학 정보 : http://ipsi.kcce.or.kr/college_info/class_human.jsp

※ 폴리텍대학 정보 : <http://ipsi.kopo.ac.kr/>

※ 특성화고 정보 : <http://www.hifive.go.kr>

2 관련학과 교과과정

중분류	소분류	교육훈련과정			
		구분	과목	내용	비율
01. 정보기술	02. 정보기술개발	대학교	SW전공기초	이산수학, 확률통계, 선형대수 등	
			프로그램 기초	프로그램 언어, 자료구조, 알고리즘 등	
			시스템SW	운영체제, 네트워크, 보안, 데이터베이스, 하드웨어 등	
			SW설계 및 응용	SW공학, SW시스템 분석	
		전문대학	프로그램 기초	프로그램 언어, 자료구조, 알고리즘 등	
			시스템SW	운영체제, 네트워크, 보안, 데이터베이스, 하드웨어 등	
		직업전문학교	기본과정	HTML5 & CSS, 프로그래밍 기초	
			심화과정	프로그래밍 고급, Database, Framework, jQuery, UML, DB설계, Mobile 프로그래밍, Web보안	
			현업요구사항	실무프로젝트	
		폴리텍 대학	스마트소프트웨어	애플리케이션, 웹 디자인, 모바일 등	
			스마트정보통신	임베디드시스템, 생산정보화시스템 구축 등	
			유비쿼터스시스템(통신)	모바일정보통신, 시스템 네트워크 기술, IT융합회로 등	
			정보통신시스템	전자, 통신, 컴퓨터 운영 기술 및 시스템 통합 등	
			컴퓨터정보	객체지향성계, 프로그래밍, 데이터베이스 관리, 운영 등	
			항공제어시스템	IT기반 항공 모션제어, 항공 임베디드 프로그래밍 등	
			IT응용제어	PC기반제어, 스마트 응용 분야	
			의료정보	병원행정관리, 의료정보관리 등 전산프로그램 개발/운영	

※ 워크넷 학과정보 / 교육통계서비스('12년)

3 자격 현황 분석

1 국가기술자격 현황

중분류	소분류	등급	종목	취득자수(명)			
				누계	'11년	'12년	'13년
01. 정보기술	02. 정보기술개발	기술사	정보관리	748	30	41	26
			컴퓨터시스템응용	516	18	30	14
			정보통신	384	29	25	21
		기사	정보처리	516,265	16,874	17,851	19,762
			전자계산기조직응용	5,156	337	215	435
			정보통신	18,343	808	329	218
			정보보안	210	-	-	210
		산업기사	정보처리	250,077	2,635	2,297	3,319
			사무자동화	349,205	12,791	12,109	10,307
			정보통신	17,631	520	219	175
			정보보안	149	-	-	149
		기능사	정보기기운용	760,922	31,241	2,270	4,652
			정보처리	619,991	5,011	5,887	13,145
		단일	게임그래픽전문가	1,801	100	42	81
		단일	게임기획전문가	1,504	134	70	79
		단일	게임프로그래밍전문가	674	22	10	17
		단일	멀티미디어콘텐츠제작전문가	2,844	217	116	122

※ 한국산업인력공단 자격정보망(Q-net)

※ 정보기술개발 소분류의 국가기술자격 등급으로는 기술사, 기사, 산업기사, 기능사 등 4개 등급, 13개 종목으로 구성되어 있으며, 누계 취득자 수는 자격시행 초기 시점부터 2013년까지의 총 누계임

※ 정보처리 종목의 기사와 기능사 자격의 경우에는 매년 취득자 수가 꾸준히 증가하고 있으며, 총 누계 인원도 타 자격에 비해 압도적으로 많은 분포를 보이고 있음

※ 총 13개 종목 중 가장 많은 취득자 수를 보이는 종목은 정보기기운용 기능사로 자격시행 초기부터 누계 취득자수가 76만여명에 이르고 있음

※ 기술사 자격은 시험 응시의 사전 제약과 시험의 고난이도를 이유로, 정보보안 관련 자격은

최근에 들어서 기술자격화 됨으로써 누계 취득자 수가 타 자격에 비해 상대적으로 적은 분포를 보이고 있음

② 국가자격 현황

중 분 류	소 분 류	종 목	등 급	취 득 자 수(명)			
				누 계	'11년	'12년	'13년
01. 정보기술	02. 정보기술개발	기술지도사	-	45	19	14	12

③ 공인민간자격 현황

중 분 류	소 분 류	종 목(등 급)	소 관 부 처	취 득 자 수(명)			
				누 계	'09년	'10년	'11년
01. 정보기술	02. 정보기술개발	데이터아키텍처전문가	미래창조과학부	10	10		
		정보시스템감리사	안전행정부	40	40		
		정보보호전문가	한국인터넷진흥원	23	23		
		PC활용능력평가 시험	피씨티	3,434	1,223	1,136	1,075
		정보시스템감리사	한국정보화진흥원	124	39	39	46
		리눅스마스터	한국정보통신진흥협회	4,771	2,042	1,799	930
		인터넷정보관리사	한국정보통신진흥협회	13,684	6,495	4,755	2,434
		디지털 정보 활용 능력 (DIAT)	한국정보통신진흥협회	230,003	44,333	103,582	82,088
		컴퓨터운용사	대한상공회의소	526	143	286	97
		PC정비사	한국정보통신자격협회	201	201		
		정보기술자격시험(ITQ)	한국생산성본부	1,091,235	502,102	522,575	66,558
		PC Master (정비사)	한국정보평가협회	383	196	187	
		정보기술프로젝트관리 전문가 (IT-PMP)	대한정보통신기술	66	21	45	

※ 한국산업인력공단 자격정보망(Q-net)

4 해외사례 분석

1 직무능력 구성

중분류	소분류	세분류(직무)	능력단위	능력단위요소
01. 정보기술	02. 정보기술개발	Solution development and implementation (솔루션 개발 및 실행)	SW 개발	감독 하에 SW개발 진행
				지정 SW개발 수행
				SW개발 운영 협조
				SW개발 프로젝트 및 프로그램 운영 협조
				SW개발 인프라 시행
				SW개발 계획
				SW개발 수행
				SW개발 업무 모니터링 및 검토
				SW개발 관리
				SW개발 인프라 운영
				SW개발 주도
				SW개발 운영
				SW개발 프로젝트 및 프로그램 운영
				SW개발 서비스 외부 공급자 관리

※ 자료 : 영국, National Occupational Standards(NOS) IT and Telecoms(2009)

※ 영국 국가직무능력표준(NOS) - 솔루션 개발 및 시행 분류 기준

※ 응용SW엔지니어링 세분류와 유사한 National Occupational Standards(NOS)의 능력단위는 SW개발 영역으로 구성되어 있으며, 능력단위 요소를 보면 개발 방법, 개발 주체, 계획, 관리 등 우리의 응용SW엔지니어링 능력단위 수준이 능력단위요소로 구성되어 있음

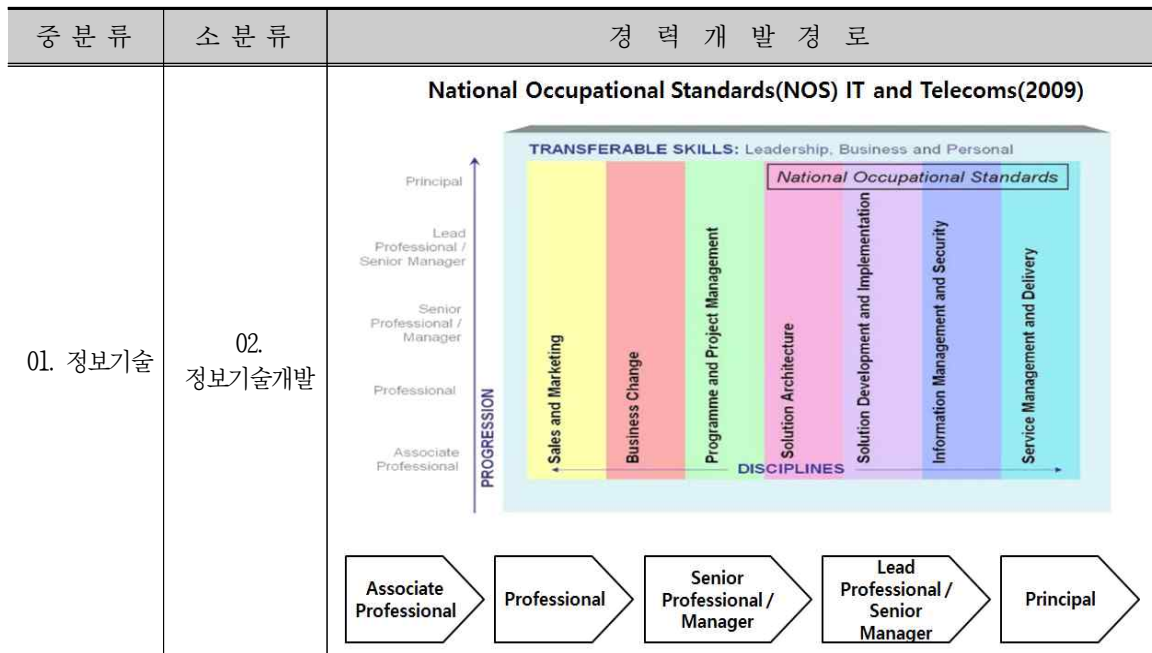
중분류	소분류	세분류(직무)	능력단위	능력단위요소
01. 정보기술	02. 정보기술개발	Solution development and implementation (솔루션 개발 및 실행)	Systems development (시스템 개발)	Systems development management (소프트웨어 개발 관리)
				Data analysis(데이터 분석)
				Systems design(시스템 설계)
				Network design(네트워크 설계)
				Database/repository design (데이터베이스/저장소 설계)
				Programming/software development (프로그래밍/소프트웨어 개발)
				Animation development(영상개발)
				Safety engineering(안전공학)
				Sustainability engineering (지속가능성 공학)
				Information content authoring (정보 콘텐츠제작)
				Testing(테스팅)
			Human factors (인적요소)	User experience analysis(사용자 경험 분석)
				Ergonomic design(시스템 인간공학 설계)
				User experience evaluation (사용자 경험 평가)
				Human factors integration(인적 요소 통합)
			Installation and integration (설치 및 통합)	Systems integration(시스템 통합)
				Porting/software integration (포팅/소프트웨어 통합)
				Systems installation/decommissioning (시스템 설치/폐기)

※ SFIA(Skill Framework for information Age)는 영국의 IT관련 주요 단체(e-skills UK, BCS, IEE, IMIS)가 연합하여 창설한 SFIA Foundation에서 개발한 “필요한 스킬의 표준모델” 제시

중분류	소분류	세분류(직무)	능력단위	능력단위요소
01. 정보기술	02. 정보기술개발	응용SW엔지니어	기능설계	화면설계
				사용자 인터페이스 디자인
				시스템관리 인터페이스 설계
				보안 설계
				정보(정보구조) 설계
				레이아웃(시각적) 설계
				데이터베이스 설계
				운영관리(서버, 회선) 디자인
				업무의 표준화
			개발 언어 및 이용 기술 설계	플랫폼 구성 설계
				장치(모바일 기기) 기술 설계
			프로그래밍(서버)	서버 사이드 프로그래밍
			프로그래밍(클라이언트)	클라이언트 사이드 프로그래밍
			테스트(디버깅)	프로그램 테스트
				시스템 통합 테스트
				사용성 테스트
				보안 테스트
			시스템 유지보수	시스템 모니터링
				서버 유지 보수

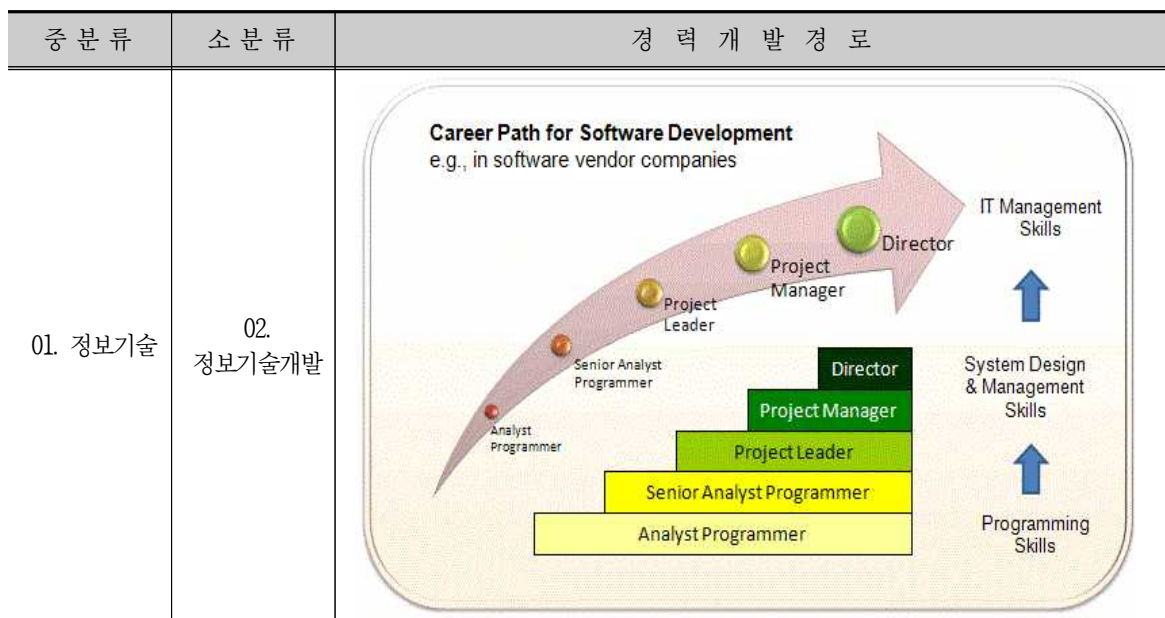
※ 일본 중앙직업능력개발협회 분류 기준

2] 경력개발경로 구성



※ 자료 : 영국, National Occupational Standards(NOS) IT and Telecoms(2009)

※ 정보기술 중분류와 유사한 영국의 NOS 체계는 솔루션아키텍처, 솔루션 개발 및 시행, 정보관리 및 보안, 서비스 관리 및 이행 등 4개 영역임



※ 자료 : 홍콩대학교

※ 홍콩대학교에서 정의하고 있는 경력개발경로는 실무 중심으로 직급과 연관성을 가져가고 있음(일반사원 - 프로젝트 리더 - 프로젝트 매니저 - 관리자)

직무명 : 응용SW엔지니어링

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	페 이 지
1	애플리케이션 요구사항 분석	21
2	요구사항 확인	29
3	기능모델 설계	34
4	정적모델 설계	39
5	동적모델 설계	44
6	인터페이스 설계	49
7	애플리케이션 설계	56
8	화면 설계	61
9	화면 구현	66
10	데이터 입출력 구현	71
11	서버프로그램 구현	77
12	인터페이스 구현	85
13	통합 구현	91
14	애플리케이션 테스트 수행	97
15	애플리케이션 테스트 관리	102
16	애플리케이션 배포	108

17	정보시스템 이행	115
18	애플리케이션 리팩토링	121
19	제품소프트웨어 패키징	126
20	프로그래밍 언어 활용	132
21	응용 SW 기초 기술 활용	138
22	소프트웨어공학 활용	144
23	소프트웨어개발 방법론 활용	149

3) 능력단위별 능력단위요소

분 류 번 호	능 력 단 위	수준	능 력 단 위 요 소
2001020219_16v4	애플리케이션 요구사항 분석	7	요구사항 도출하기
			요구사항 분석하기
			요구사항 명세화하기
			요구사항 검증하기
2001020201_16v3	요구사항 확인	5	현행 시스템 분석하기
			요구사항 확인하기
			분석모델 확인하기
2001020220_16v1	기능모델 설계	6	기능 모델 작성하기
			기능 모델 검증하기
2001020222_16v4	정적모델 설계	6	정적 분석모델 검증하기
			정적모델 상세화하기
2001020223_16v4	동적모델 설계	6	동적 분석모델 검증하기
			동적모델 상세화하기
2001020218_16v4	인터페이스 설계	6	인터페이스 요구사항 확인하기
			인터페이스 대상 식별하기
			인터페이스 상세 설계하기
2001020221_16v4	애플리케이션 설계	6	공통 모듈 설계하기
			타 시스템 연동설계하기
2001020224_16v4	화면 설계	5	UI 요구사항 확인하기
			UI 설계하기
2001020225_16v4	화면 구현	3	UI 설계 확인하기
			UI 구현하기
2001020205_16v4	데이터 입출력 구현	5	논리 데이터저장소 확인하기
			물리 데이터저장소 설계하기
			데이터 조작 프로시저 작성하기
			데이터 조작 프로시저 최적화하기
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
2001020206_16v4	통합 구현	5	연계 데이터 구성하기
			연계 매카니즘 구성하기

			내외부 연계 모듈 구현하기
2001020227_16v4	애플리케이션 테스트 수행	3	애플리케이션 테스트 수행하기
			애플리케이션 결함 조치하기
2001020226_16v4	애플리케이션 테스트 관리	5	애플리케이션 테스트케이스 설계하기
			애플리케이션 통합 테스트하기
			애플리케이션 성능 개선하기
2001020214_16v4	애플리케이션 배포	3	애플리케이션 배포 환경 구성하기
			애플리케이션 소스 검증하기
			애플리케이션 빌드하기
			애플리케이션 배포하기
2001020208_16v3	정보시스템 이행	5	정보시스템 사용자 교육하기
			정보시스템 이행하기
			정보시스템 안정화하기
2001020217_16v4	애플리케이션 리팩토링	6	리팩토링 기준 수립하기
			소스 코드 분석하기
			리팩토링 적용하기
2001020209_16v4	제품소프트웨어 패키징	5	제품소프트웨어 패키징하기
			제품소프트웨어 매뉴얼 작성하기
			제품소프트웨어 버전관리하기
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
2001020228_16v4	소프트웨어공학 활용	6	CASE 도구 활용하기
			품질요구사항 확인하기
2001020229_16v4	소프트웨어개발 방법론 활용	6	소프트웨어개발 방법론 선정하기
			소프트웨어개발 방법론 테일러링하기

2. 능력단위별 세부내용

분류번호 : 2001020219_16v4

능력단위 명칭 : 애플리케이션 요구사항 분석

능력단위 정의 : 애플리케이션 요구사항 분석이란 구현하고자 하는 애플리케이션의 요구사항을 도출, 분석, 명세화 및 요구사항 검증을 수행하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020219_16v4.1 요구사항 도출하기	<p>1.1 비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구사항 도출을 위한 정보를 수집 할 수 있다.</p> <p>1.2 인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다.</p> <p>1.3 요구사항 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.</p>
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ SW 제품 개발 수명 주기 이론 ○ 요구 공학(Requirement Engineering) 이론 ○ SW 제품 요구사항 관리 절차 ○ SW 제품 요구사항 도출 지식 ○ SW 제품 요구사항 정의 기준 ○ 요구사항 명세서(Software Requirement Specification) 작성 기준 ○ IEEE830 (IEEE Recommended Practice for Software Requirements Specifications) ○ 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 ○ BABOK (Business Analysis Body of Knowledge) ○ SWEBOK(Software Engineering Body of Knowledge) <p>【기 술】</p> <ul style="list-style-type: none"> ○ 요구사항 도출 기법 ○ 인터뷰 기법 ○ 시나리오 작성 기법 ○ 프로토타입 제작 기법 ○ 그룹 회의 토의 진행 기법 ○ 사용자 스토리 작성 기법 ○ 요구사항 기록 방법 <p>【태 도】</p>

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 도출한 요구사항을 사용자 관점에서 정확하게 이해하고자 하는 태도 ○ 다양한 출처로 부터의 요구사항을 명확하게 도출하고자 노력하는 태도 ○ 요구사항 도출 후 정확하게 기록하고자 하는 태도 ○ 구체적인 요구사항 도출을 위해 다양한 기법을 활용하고자 하는 열린 자세
<p>2001020219_16v4.2 요구사항 분석하기</p>	<p>2.1 요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구사항과 비기능적 요구사항을 분류 할 수 있다.</p> <p>2.2 도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다.</p> <p>2.3 도출한 요구사항이 개발하고자 하는 애플리케이션 전체 또는 단위 컴포넌트에 향해 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다.</p> <p>2.4 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선 순위, 영향 범위 등을 확정할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ SW제품 개발 요구 공학 ○ SW제품 품질 속성 항목 ○ SW제품 품질 평가 항목 ○ 정보시스템의 기술적 지식 ○ 요구사항 적용 범위 확인 지식 ○ 요구사항 우선 순위 결정 기준 ○ 요구사항 영향범위 설정 기준 ○ SW 제품 개발 수명 주기 이론 ○ 요구 공학(Requirement Engineering) 이론 ○ SW 제품 요구사항 관리 절차 ○ SW 제품 요구사항 정의 기준 ○ 요구사항 명세서(Software Requirement Specification) 작성 기준 ○ IEEE830 (IEEE Recommended Practice for Software Requirements Specifications) ○ 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 ○ BABOK (Business Analysis Body of Knowledge) ○ SWEBOK(Software Engineering Body of Knowledge) <p>【기 술】</p> <ul style="list-style-type: none"> ○ 요구사항 분석 능력 ○ 요구사항 분석 기술 ○ 요구사항 분류 기술 ○ 요구사항의 우선순위를 식별하는 능력 ○ 사용자와 협상 능력 ○ 상위레벨 SW제품 아키텍처 수립 기술

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 품질 속성 Trade-off 방법 ○ SW제품 요구사항 변경 및 추적 기술 ○ 비기능 요구사항 식별 기술 【태 도】 <ul style="list-style-type: none"> ○ 고객관점의 요구사항을 이해하려는 태도 ○ 요구 분석 내용을 체계적으로 이해하고자 하는 태도 ○ 요구사항을 명확하게 하기 위해 사용자와 협상하는 태도 ○ 요구사항 분석 기준을 통해 우선순위를 결정하는 태도 ○ 요구사항 영향범위를 명확히하고자 하는 태도
2001020219_16v4.3 요구사항 명세화하기	<p>3.1 애플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의서를 작성할 수 있다.</p> <p>3.2 구현하고자 하는 애플리케이션이 구동되는 시스템이나 하드웨어 관련 내용을 구체화 하여 시스템 요구사항 문서를 작성할 수 있다.</p> <p>3.3 구현하고자 하는 애플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)을 작성할 수 있다.</p> 【지 식】 <ul style="list-style-type: none"> ○ SW 제품 개발 수명 주기 이론 ○ 요구 공학(Requirement Engineering) 이론 ○ SW 제품 요구사항 관리 절차 ○ SW 제품 요구사항 정의 기준 ○ 요구사항 명세서(Software Requirement Specification) 작성 기준 ○ IEEE830 (IEEE Recommended Practice for Software Requirements Specifications) ○ 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 ○ BABOK (Business Analysis Body of Knowledge) ○ SWEBOK(Software Engineering Body of Knowledge) 【기 술】 <ul style="list-style-type: none"> ○ 요구사항 명세 기술 ○ 수집된 자료에 대한 체계적인 분류 및 정리 기술 ○ 요구사항 정의서 문서 작성 능력 ○ 이해당사자와 커뮤니케이션 능력 ○ 인터뷰 기술 【태 도】 <ul style="list-style-type: none"> ○ 요구사항 정의서 양식에 맞추어 체계적으로 작성하고자 하는 태도 ○ 개발 제품의 요건을 명확하게 도출 할 수 있는 태도

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 구체적이고 명확한 요구사항 정의서를 작성하고자 하는 태도 ○ 요구사항 명세서의 내용을 고객과 개발팀이 이해할 수 있도록 기술하는 노력
2001020219_16v4.4 요구사항 검증하기	<p>4.1 소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다.</p> <p>4.2 작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한 설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다.</p> <p>4.3 기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다.</p> <p>4.4 분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ SW 제품 개발 수명 주기 이론 ○ 요구 공학(Requirement Engineering) 이론 ○ SW 제품 요구사항 관리 절차 ○ IEEE830 (IEEE Recommended Practice for Software Requirements Specifications) ○ 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 ○ BABOK (Business Analysis Body of Knowledge) ○ SWEBOOK(Software Engineering Body of Knowledge) <p>【기 술】</p> <ul style="list-style-type: none"> ○ 요구사항 검증 기법 ○ 요구사항 명세서 조사 기법 ○ 요구사항 명세서 검토 기법 ○ 요구사항 검증 기준 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 요구사항 명세서의 일관성과 완성도를 검증하고자 하는 자세 ○ 요구사항 명세서의 오류를 발견하여 조치하고자 하는 태도 ○ 요구사항의 구현 가능성, 안정성, 기술적 환경 분석 자세 ○ 사용자 요구사항 반영 여부를 확인하고자 하는 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 응용SW엔지니어링 애플리케이션 요구사항 분석 업무에 적용한다.
- 다양한 요구사항을 이해하고 분석하여 어플리케이션 설계에 필요한 요구사항에 대한 기준(분류, 우선 순위, 영향범위 등)을 수립하고 명세화하여야 한다.
- 이 능력단위는 사용자와의 대화, 협의 및 협상을 할 수 있는 기법에 대한 이해와 하드웨어 시스템에 대한 이해가 선행되어야 한다.
- 이 능력단위의 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련에 적용한다.
- 이 능력단위를 적용하여 응용소프트웨어 개발의 타당성 여부를 검토하고, 이를 바탕으로 개발 기준 제시를 통해 응용소프트웨어 개발하는데 결함의 가능성을 줄일 수 있다.
- 요구사항 검증하기 능력단위 요소는 작성한 요구사항 명세가 도출된 요구사항을 반영하여 요구사항 분석대로 작성되었는지 조사(Inspection), 검토(Review)를 수행하는 것이 목적이며, “요구사항 확인” 능력단위는 애플리케이션 설계를 위해 검증된 요구사항 명세를 확인하는 것이다.

자료 및 관련 서류

- 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)
- 요구사항 정의서
- IEEE SW Engineering 표준 산출물
- IEEE 830 (IEEE Recommended Practice for Software Requirements Specifications)
- 정보시스템 마스터플랜(ISMP, Information System Master Plan) 방법론
- SW 사업 대가 산정 가이드
- IFPUG Function Point Counting Practices Manual
- ISO/IEC 15504(Information Technology Software Process Assessment)
- ISO/IEC 9126 (Information Technology Software Quality Characteristics and Metrics)
- ISO/IEC 14598 (Information Technology Software Product Evaluation)

장비 및 도구

- 전산장비 : 인터넷, 컴퓨터, 프린터, 복사기, 빔 프로젝터, 등
- 지원도구 : 통계 분석도구, 문서작성 도구 등
- OA소프트웨어 (워드프로세서, 스프레드시트, 프레젠테이션 도구 등)

재료

- 해당없음

● 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 요구사항 분석의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오 (윤고덕110, 10p, 90%, -10)	√	√
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험	√	√
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트		
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

- 요구사항 도출 기법에 대한 이해
- 사용자 요구사항 정의서 작성 방법
- 사용되는 시스템이나 하드웨어에 대한 이해
- 사용되는 소프트웨어에 대한 이해
- 소프트웨어 요구사항 명세서 작성 방법에 대한 이해
- 도출한 요구사항에 대한 기준 수립에 대한 이해
- 요구사항 명세서에 대한 검증 및 검토에 대한 이해

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020219_15v3
	보완	2001020219_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020201_16v3

능력단위 명칭 : 요구사항 확인

능력단위 정의 : 요구사항 확인이란 업무 분석가가 수집·분석·정의한 요구사항과 이에 따른 분석모델에 대해서 확인과 현행 시스템에 대해 분석하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020201_16v3.1 현행 시스템 분석하기	<p>1.1 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다.</p> <p>1.2 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다.</p> <p>1.3 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 해당 플랫폼에 대한 지식○ 프로젝트 환경 및 특수성○ 플랫폼에 따른 기능 및 성능 특성○ 가상화 관련 지식○ 클라우드 컴퓨팅 관련 지식 <p>【기 술】</p> <ul style="list-style-type: none">○ 내부 및 외부 환경 분석 기술○ 운영체제 구성 및 관리 능력○ 저장장치 구성 및 관리 능력○ 네트워크 구성 및 관리 능력○ DBMS 구성 및 관리 기술○ 가상화 관련 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 기술 관련 각종 정보 수집에 대한 적극성○ 용어정의 정확성과 완전성을 기하고자 하는 의지○ 분류 및 정리 태도○ 시스템에 대한 정확성과 이해의 완전성을 갖고자 하는 태도○ 시스템과 개발 소프트웨어와의 관계를 파악하려는 태도
2001020201_16v3.2 요구사항 확인하기	<p>2.1 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다.</p> <p>2.2 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다.</p> <p>2.3 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 프로젝트 환경 및 특수성○ 요구공학 방법론

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 요구분석기법 ○ 소프트웨어 개발 방법론 ○ 타당성 분석기법 ○ 통계학 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 유즈케이스 작성 능력 ○ UML 작성 기술 ○ 분석 자동화도구 도구 사용 능력 ○ 요구사항 관리 도구 사용 기술 ○ 리뷰 진행 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 요구사항의 정확성과 완전성을 확보하려는 자세 ○ 정확한 유즈케이스를 이해하고 분석하려는 자세 ○ 검증할 항목 분석을 위한 치밀한 태도 ○ 책임감 및 검증에 대한 완벽함을 추구하는 태도
<p>2001020201_16v3.3 분석모델 확인하기</p>	<p>3.1 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다.</p> <p>3.2 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다.</p> <p>3.3 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 프로젝트 환경 및 특수성 ○ 요구공학 방법론 ○ 모델링 기법 ○ 요구분석기법 ○ 용어사전 작성 방법 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 유즈케이스 작성 능력 ○ UML 작성 기술 ○ 분석 자동화도구 도구 활용 기술 ○ 요구사항 관리 도구 사용 기술 ○ 리뷰 진행 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객의 요청을 수용하고자하는 태도 ○ 용어정의 정확성과 완전성을 기하고자 하는 의지 ○ 요구사항의 정확성과 완전성을 확보하려는 자세 ○ 요구사항을 정확하게 기술하려는 자세 ○ 비즈니스의 내용을 정확히 반영하는 프로세스 모델을 만들려는 치밀한 자세 ○ 요구사항의 명확성을 찾으려는 자세 ○ 요구사항에 대한 일관성을 부여하는 자세 ○ 요구사항을 유기적 관계를 발견하는 자세

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위의 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련에 적용한다.
- 이 능력단위는 업무 분석가가 분석한 요구사항 및 분석모델에 대해서, 응용소프트웨어개발에 필요한 내용을 확인하는데 적용된다.
- 이 능력단위는 응용소프트웨어 개발을 위한 시스템의 설계에 활용된다.
- 이 능력단위는 응용소프트웨어 개발의 타당성 여부를 검토하는데 활용되서는 안된다. 이 능력단위를 수행하기 이전에 시스템 개발의 타당성 검토는 이미 이루어져야 한다.
- 이 능력단위를 적용하여 응용소프트웨어를 개발하게 되면 분석과 설계가 매끄럽게 연결될 수 있기 때문에 단계의 이동에 따른 결함의 가능성을 줄일 수 있다.
- 요구사항 검증하기 능력단위 요소는 작성한 요구사항 명세가 도출된 요구사항을 반영하여 요구사항 분석대로 작성되었는지 조사(Inspection), 검토(Review)를 수행하는 것이 목적이며, “요구사항 확인” 능력단위는 애플리케이션 설계를 위해 검증된 요구사항 명세를 확인하는 것이다.

자료 및 관련 서류

- 각 제조사 운영체제의 시스템 및 네트워크 관리자 매뉴얼
- 각 제조사 보안 장비의 관리 및 운영 매뉴얼
- 정보시스템 하드웨어 규모산정지침(TTAK.KO-10.0292)
- ISO 27001
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 모델링 검토기준
- 작업 일지

장비 및 도구

- 각 제조사 별 서버 관리 도구
- IT자산관리 시스템
- SMS (Server Management System)
- 컴퓨터, 프린터, 인터넷
- 형상관리 도구
- 모델링 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 요구사항 확인의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 산업 분야에 대한 분석
 - 요구사항 명세화의 정확성 여부

- 서버 하드웨어 용량 산정 방법 숙지 및 수행 내역
- 운영체제의 주요 기능, 구조, 특성에 대한 분석
- 미들웨어(DBMS/WAS)의 주요 기능, 구조, 특성에 대한 분석

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀워크능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
5	정보능력	컴퓨터활용능력, 정보처리능력

◎ 개발 이력(신규양식)

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020201_14v2
	보완	2001020201_16v3
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v3
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020220_16v1

능력단위 명칭 : 기능모델 설계

능력단위 정의 : 기능모델 설계란 애플리케이션 요구사항에 의해 도출, 분석된 내용을 애플리케이션의 기능으로 명세화하여 요구사항을 검증하고 소프트웨어 개발범위를 확정하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020220_16v1.1 기능모델 작성하기	<p>1.1 기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다.</p> <p>1.2 기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별하여 다이어그램을 작성할 수 있다.</p> <p>1.3 기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능모델 설계 명세서를 작성할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 요구 공학(Requirement Engineering) 이론○ SWEBOK(Software Engineering Body of Knowledge) <p>【기 술】</p> <ul style="list-style-type: none">○ UML 작성 기술○ 모델링 기술○ 모델링 명세화 기술○ 모델 검증 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 고객의 요청을 적극적으로 수용하고자 하는 의지○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도○ 기능모델의 일관성과 완성도를 검증하고자 하는 자세○ 기능의 누락이나 오류를 발견하여 조치하고자 하는 태도○ 산출물 완성도 확보를 위한 적극적인 태도○ 고객과의 지속적인 소통을 위한 책임감 있는 자세
2001020220_16v1.2 기능모델 검증하기	<p>2.1 도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다.</p> <p>2.2 작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다.</p> <p>2.3 작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함하고 있는지 검증할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 개발 방법론 이해 ○ 요구 공학(Requirement Engineering) 이론 ○ SWEBOK(Software Engineering Body of Knowledge) 【기 술】 ○ UML 작성 기술 ○ 모델링 기술 ○ 모델 검증 기술 ○ 의사소통 기술 【태 도】 ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 ○ 기능모델의 일관성과 완성도를 검증하고자 하는 자세 ○ 기능의 누락이나 오류를 발견하여 조치하고자 하는 태도 ○ 산출물 완성도 확보를 위한 적극적인 태도 ○ 고객과의 지속적인 소통을 위한 책임감 있는 태도

◎ 적용범위

고려사항

- 이 능력단위는 응용SW엔지니어링 기능모델 설계 업무에 적용한다.
- 다양한 요구사항을 이해하고 분석하여 기능모델 설계에 필요한 요구사항에 대한 기준(분류, 우선순위, 영향범위 등)을 수립하고 명세화하여야 한다.
- 이 능력단위는 사용자와의 대화, 협의 및 협상을 할 수 있는 기법에 대한 이해와 하드웨어 시스템에 대한 이해가 선행되어야 한다.
- 이 능력단위의 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련에 적용한다.

자료 및 관련 서류

- 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)
- 요구사항 정의서
- IEEE SW Engineering 표준 산출물
- IEEE 830 (IEEE Recommended Practice for Software Requirements Specifications)
- 정보시스템 마스터플랜(ISMP, Information System Master Plan) 방법론
- SW 사업 대가 산정 가이드
- IFPUG Function Point Counting Practices Manual
- ISO/IEC 15504(Information Technology Software Process Assessment)
- ISO/IEC 9126 (Information Technology Software Quality Characteristics and Metrics)
- ISO/IEC 14598 (Information Technology Software Product Evaluation)

장비 및 도구

- 전산장비 : 인터넷, 컴퓨터, 프린터, 빔 프로젝터 등
- 지원도구 : 문서작성 도구 등
- OA소프트웨어 (워드프로세서, 스프레드시트, 프레젠테이션 도구 등)

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 기능모델 설계의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오 (운고덕110, 10p, 90%, -10)	√	√
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험	√	√
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트		
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 요구사항 도출 기법에 대한 이해
 - 사용자 요구사항 정의서 작성 방법
 - 사용되는 시스템이나 하드웨어에 대한 이해
 - 사용되는 소프트웨어에 대한 이해
 - 소프트웨어 요구사항 명세서 작성 방법에 대한 이해

- 도출한 요구사항에 대한 기준 수립에 대한 이해
- 요구사항 명세서에 대한 검증 및 검토에 대한 이해

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		신설
분류번호	기존	-
	보완	2001020220_16v1
개발연도	현재	2016
	최초(1차)	2016
버전번호		v1
개발자	현재	한국소프트웨어산업협회
	최초(1차)	한국소프트웨어산업협회
향후 보완 연도(예정)		2019-2021
능력단위 보완사유		

분류번호 : 2001020222_16v4

능력단위 명칭 : 정적모델 설계

능력단위 정의 : 정적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 정적모델을 설계하고 검증하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020222_16v4.1 정적 분석모델 검증하기	<p>1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내역을 확인할 수 있다.</p> <p>1.2 분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 소프트웨어 아키텍처에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none">○ UML 작성 기술○ 모델링 기술○ 모델 검증 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 고객의 요청을 적극적으로 수용하고자 하는 태도○ 책임감 및 설계문서에 완벽함을 추구하는 태도○ 아키텍처와 설계자간의 협업을 추구하는 태도○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도○ 산출물 완성도를 위한 적극적인 태도○ 고객과의 지속적인 소통을 위한 책임감 있는 태도○ 작업환경 개선 의지
2001020222_16v4.2 정적모델 상세화하기	<p>2.1 분석된 정적모델을 정적 설계모델로 변환하기 위해 변환규칙을 정의할 수 있다.</p> <p>2.2 정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다.</p> <p>2.3 상세화된 정적 설계모델의 명세서를 작성할 수 있다.</p> <p>2.4 설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 업무 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 소프트웨어 아키텍처에 대한 이해○ 설계 모델링 기법○ 데이터베이스 이해○ 네트워크 이해 <p>【기 술】</p> <ul style="list-style-type: none">○ UML 작성 기술

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 설계 모델링 기술 ○ E-R 모델 작성 기술 ○ IDE 및 개발환경 도구 활용 기술 ○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 기술 ○ 기술영역별 미들웨어/솔루션 활용 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 책임감 및 설계문서에 완벽함을 추구하는 태도 ○ 아키텍트와 설계자간의 협업을 추구하는 태도 ○ 도출된 모듈 및 컴포넌트를 분석하기 위한 치밀한 태도 ○ 산출물 완성도 확보를 위한 적극적인 태도 ○ 주어진 과제를 완수하는 책임감

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 수작업이 아닌 도구를 활용한 상세설계와 요구사항의 연계를 고려해야 한다.
- 이 능력단위는 소프트웨어 아키텍처 변동에 따른 유연성을 확보해야 한다.
- 이 능력단위에 사용되는 기술은 자주 변경되기 때문에 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련이 요구된다.
- 이 능력단위는 여러 개발방법론에 대한 이해가 우선되어야 한다.
- 훈련시간을 고려하여 애플리케이션 설계에 공통 모듈설계와 타 시스템 연동 설계로 수정하고 정적모델 설계와 동적모델 설계로 분할한다.

자료 및 관련 서류

- UML 작성표준
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 모델링 검토기준
- 작업일지
- 공공부문 데이터 모델링 언어 적용 지침

장비 및 도구

- 컴퓨터, 프린터
- 인터넷
- 설계용 소프트웨어

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 정적모델 설계의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 요구사항 추적성 확보
 - 소프트웨어 아키텍처와 일관성 확인
 - 프로토타입 작성 및 검증 능력

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀워크능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
5	정보능력	컴퓨터활용능력, 정보처리능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020202_15v3
	보완	2001020221_16v4, 2001020222_16v4, 200102023_16v4,
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020223_16v4

능력단위 명칭 : 동적모델 설계

능력단위 정의 : 동적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 동적모델을 설계하고 검증하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020223_16v4.1 동적 분석모델 검증하기	<p>1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다.</p> <p>1.2 분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 소프트웨어 아키텍처에 대한 이해○ 설계 모델링 기법○ 네트워크 이해 <p>【기 술】</p> <ul style="list-style-type: none">○ UML 작성 기술○ 설계 모델링 기술○ E-R 모델 작성 기술○ IDE 및 개발환경 도구 활용○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용○ 기술영역별 미들웨어/솔루션 활용 <p>【태 도】</p> <ul style="list-style-type: none">○ 고객의 요청을 적극적으로 수용하고자 하는 태도○ 책임감 및 설계문서에 완벽함을 추구하는 태도○ 아키텍트와 설계자간의 협업을 추구하는 태도○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도○ 산출물 완성도를 위한 적극적인 태도○ 주어진 과제를 완수하는 책임감
2001020223_16v4.2 동적모델 상세화 하기	<p>2.1 검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다.</p> <p>2.2 동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다.</p> <p>2.3 상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다.</p> <p>2.4 도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 소프트웨어 아키텍처에 대한 이해

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 설계 모델링 기법 ○ 네트워크 이해 ○ 분산 프로그래밍에 대한 이해 【기 술】 ○ UML 작성 기술 ○ 설계 모델링 기술 ○ E-R 모델 작성 기술 ○ IDE 및 개발환경 도구 활용 기술 ○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 기술 ○ 기술영역별 미들웨어/솔루션 활용 【태 도】 ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 책임감 및 설계문서에 완벽함을 추구하는 태도 ○ 아키텍트와 설계자간의 협업을 추구하는 태도 ○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 ○ 산출물 완성도를 위한 적극적인 태도 ○ 주어진 과제를 완수하는 책임감

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 수작업이 아닌 도구를 활용한 상세설계와 요구사항의 연계를 고려해야 한다.
- 이 능력단위는 소프트웨어 아키텍처 변동에 따른 유연성을 확보해야 한다.
- 이 능력단위에 사용되는 기술은 자주 변경되기 때문에 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련이 요구된다.
- 이 능력단위는 여러 개발방법론에 대한 이해가 우선되어야 한다.
- 훈련시간을 고려하여 애플리케이션 설계에 공통 모듈설계와 타 시스템 연동 설계로 수정하고 정적모델 설계와 동적모델 설계로 분할한다.

자료 및 관련 서류

- UML 작성표준
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 모델링 검토기준
- 작업일지
- 공공부문 데이터 모델링 언어 적용 지침

장비 및 도구

- 컴퓨터, 프린터, 복사기, 계산기
- 인터넷
- 설계용 소프트웨어

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 동적모델 설계의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 요구사항 추적성 확보
 - 소프트웨어 아키텍처와 일관성 확인
 - 프로토타입 작성 및 검증 능력

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀워크능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
5	정보능력	컴퓨터활용능력, 정보처리능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020202_15v3
	보완	2001020221_16v4, 2001020222_16v4, 200102023_16v4,
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020218_16v4

능력단위 명칭 : 인터페이스 설계

능력단위 정의 : 인터페이스 설계란 응용소프트웨어 개발을 위해 정의된 시스템 인터페이스 요구사항을 확인하고 인터페이스 대상을 식별하여 인터페이스 설계서를 작성하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020218_16v4.1 인터페이스 요구사항 확인하기	<p>1.1 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용 소프트웨어의 인터페이스 비 기능 요구사항을 분석할 수 있다.</p> <p>1.2 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용 소프트웨어의 인터페이스 기능 요구사항을 분석할 수 있다.</p> <p>1.3 개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의 정확성과 완전성을 확인한다.</p> <p>【지 식】</p> <ul style="list-style-type: none">◦ 산업 분야별 업무특성◦ 요구공학◦ 소프트웨어 아키텍처◦ 네트워크◦ 설계 모델링 방법론 <p>【기 술】</p> <ul style="list-style-type: none">◦ UML 작성 기술◦ 설계 모델링 기술◦ 요구사항 관리 도구 사용 능력◦ IDE 및 개발환경 도구 활용◦ 리뷰 진행 기술 <p>【태 도】</p> <ul style="list-style-type: none">◦ 정확한 요구사항의 이해를 위한 분석 자세◦ 이해 당사자와의 협업을 위한 적극적 의사소통◦ 조직의 개발방법론을 준수하려는 태도◦ 조직의 품질관리 기준을 준수하려는 태도
2001020218_16v4.2 인터페이스 대상 식별하기	<p>2.1 개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대상 시스템을 식별할 수 있다.</p> <p>2.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 연계 시스템을 식별할 수 있다.</p> <p>2.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 송수신 데이터를 식별할 수 있다.</p> <p>【지 식】</p>

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 산업 분야별 업무특성 ○ 요구공학 ○ 소프트웨어 아키텍처 ○ 시스템 아키텍처 ○ 네트워크 ○ 인터페이스 시스템 【기 술】 ○ UML 작성 기술 ○ 요구사항 관리 도구 사용 능력 ○ IDE 및 개발환경 도구 활용 ○ 기술영역별 미들웨어와 솔루션 활용 능력 ○ 프레임워크 활용 【태 도】 ○ 정확한 인터페이스 시스템 식별을 위한 의지 ○ 이해 당사자와의 협업을 위한 적극적 의사소통 ○ 조직의 개발방법론을 준수하려는 태도 ○ 조직의 품질관리 기준을 준수하려는 태도 ○ 산출물 완성도 확보를 위한 적극적인 자세
2001020218_16v4.3 인터페이스 상세 설계하기	<p>3.1 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위한 송수신 방법을 명세화할 수 있다.</p> <p>3.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 데이터를 명세화할 수 있다.</p> <p>3.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오류 시 처리방안을 명세화할 수 있다.</p> <p>3.4 소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성할 수 있다.</p> 【지 식】 <ul style="list-style-type: none"> ○ 산업 분야별 업무특성 ○ 요구공학 ○ 소프트웨어 아키텍처 ○ 네트워크 ○ 설계 모델링 방법론 【기 술】 <ul style="list-style-type: none"> ○ UML 작성 기술 ○ 설계 모델링 기술 ○ IDE 및 개발환경 도구 활용 ○ 기술영역별 미들웨어와 솔루션 활용 능력 ○ 프레임워크 활용

능 력 단 위 요 소	수 행 준 거
	<p>【태 도】</p> <ul style="list-style-type: none"> ○ 사용자의 요구사항에 대한 적극적 수용 태도 ○ 이해 당사자와의 협업을 위한 적극적 의사소통 ○ 조직의 개발방법론을 준수하려는 태도 ○ 조직의 품질관리 기준을 준수하려는 태도 ○ 산출물 완성도 확보를 위한 적극적인 자세

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위의 적절한 수행을 위해서 SW아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 반드시 참조한다.
- 이 능력단위는 정적모델 상세설계하기에서 상세화환 제어 클래스 또는 사용자와 시스템 간 상호작용에 관련된 제어관계를 참조하여 수행한다.
- 이 능력단위는 동적모델 설계의 상태차트, 시퀀스 다이어그램, 협동 다이어그램을 통한 유즈케이스상세화를 참조하여 수행한다.
- 이 능력단위는 여러 개발방법론에 대한 이해가 선행되어야 하며 소프트웨어 아키텍처 변동에 따른 유연성을 확보해야 한다.

자료 및 관련 서류

- 대내/외 인터페이스 표준 및 설계지침
- 연계 데이터 표준 및 설계지침
- UML 작성표준
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 모델링 검토기준
- 작업 일지

장비 및 도구

- IT자산관리 시스템
- 컴퓨터, 프린터, 인터넷
- 형상관리 도구
- 모델링 도구
- 설계용 소프트웨어

자료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 인터페이스 설계하기의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 인터페이스 요구사항 분석 및 확인 절차
 - 인터페이스 대상 시스템 및 연계 시스템 식별 과정
 - 인터페이스 대상 송수신 데이터 식별 및 구성
 - 내외부 시스템간 인터페이스 설계서 작성 절차
 - 소프트웨어 아키텍처와 일관성 여부

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020218_15v3
	보완	2001020218_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020221_16v4

능력단위 명칭 : 애플리케이션 설계

능력단위 정의 : 애플리케이션 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통모듈 설계, 타 시스템 연동에 대하여 상세 설계하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020221_16v4.1 공통 모듈 설계하기	<div>1.1 재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다.</div> <div>1.2 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다.</div> <div>1.3 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다.</div> <div>1.4 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.</div> <div>【지 식】<ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 개발 방법론 이해○ 소프트웨어 아키텍처에 대한 이해○ 설계 모델링 기법○ 네트워크 이해○ 분산 프로그래밍에 대한 이해【기 술】<ul style="list-style-type: none">○ UML 작성 기술○ 설계 모델링 기술○ E-R 모델 작성 기술○ IDE 및 개발환경 도구 활용○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용○ 기술영역별 미들웨어/솔루션 활용【태 도】<ul style="list-style-type: none">○ 고객의 요청을 적극적으로 수용하고자 하는 태도○ 책임감 및 설계문서에 완벽함을 추구하는 태도○ 아키텍트와 설계자간의 협업을 추구하는 태도○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도○ 산출물 완성도를 위한 적극적인 태도○ 주어진 과제를 완수하는 책임감</div>
2001020221_16v4.2 타 시스템 연동설계하기	<div>2.1 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다.</div> <div>2.2 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다.</div> <div>2.3 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어/</div>

능 력 단 위 요 소	수 행 준 거
	<p>솔루션에 대하여 명세를 작성할 수 있다.</p> <p>2.4 소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 개발 방법론 이해 ○ 소프트웨어 아키텍처에 대한 이해 ○ 설계 모델링 기법 ○ 네트워크 이해 ○ 분산 프로그래밍에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ UML 작성 기술 ○ 설계 모델링 기술 ○ E-R 모델 작성 기술 ○ IDE 및 개발환경 도구 활용 ○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 ○ 기술영역별 미들웨어/솔루션 활용 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 책임감 및 설계문서에 완벽함을 추구하는 태도 ○ 아키텍트와 설계자간의 협업을 추구하는 태도 ○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 ○ 산출물 완성도를 위한 적극적인 태도 ○ 주어진 과제를 완수하는 책임감

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 수작업이 아닌 도구를 활용한 상세설계와 요구사항의 연계를 고려해야 한다.
- 이 능력단위는 소프트웨어 아키텍처 변동에 따른 유연성을 확보해야 한다.
- 이 능력단위에 사용되는 기술은 자주 변경되기 때문에 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련이 요구된다.
- 이 능력단위는 여러 개발방법론에 대한 이해가 우선되어야 한다.
- 이 능력단위는 훈련시간을 고려하여 애플리케이션 설계에 공통 모듈설계와 타 시스템 연동 설계로 수정하고 정적모델 설계와 동적모델 설계로 분할한다.

자료 및 관련 서류

- UML 작성표준
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 모델링 검토기준
- 작업일지
- 공공부문 데이터 모델링 언어 적용 지침

장비 및 도구

- 컴퓨터, 프린터, 복사기, 계산기
- 인터넷
- 설계용 소프트웨어

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 설계의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오		
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 요구사항 추적성 확보
 - 소프트웨어 아키텍처와 일관성 확인
 - 프로토타입 작성 및 검증 능력

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀워크능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
5	정보능력	컴퓨터활용능력, 정보처리능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020202_15v3
	보완	2001020221_16v4, 2001020222_16v4, 200102023_16v4,
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020224_16v4

능력단위 명칭 : 화면 설계

능력단위 정의 : 화면 설계란 요구사항분석 단계에서 파악된 화면에 대한 요구사항을 소프트웨어 아키텍처 단계에서 정의된 구현 지침 및 UI/UX 엔지니어가 제시한 UI표준과 지침에 따라 화면을 설계하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020224_16v4.1 UI 요구사항 확인하기	<div>1.1 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.</div> <div>1.2 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다.</div> <div>1.3 작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.</div> <div>【지 식】<ul style="list-style-type: none">○ 구현에 요구되는 아키텍처의 이해○ 유용성 개념 및 적용원리○ UI 기획/설계/구현 절차○ UI 표준 및 지침○ 유즈케이스에 대한 이해○ 데이터 타입 특성○ 프로토타입 작성법【기 술】<ul style="list-style-type: none">○ 유즈케이스 작성 능력○ 스토리보드 작성 기술○ 프로토타입 도구 사용 능력○ IDE 도구 활용 능력○ UI 작성 그래픽 도구 활용 능력【태 도】<ul style="list-style-type: none">○ 고객을 적극적으로 이해하고 수용하기 위한 자세○ UI 표준 및 지침을 준수하려는 태도○ UI 관련자들과 원활한 관계를 유지하려는 태도○ 기술적 위험에 적극적으로 대비하려는 노력</div>
2001020224_16v4.2 UI 설계하기	<div>2.1 UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼의 흐름을 설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다.</div> <div>2.2 UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다.</div> <div>2.3 UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.</div> <div>【지 식】<ul style="list-style-type: none">○ 구현에 요구되는 아키텍처의 이해○ 유용성 개념 및 적용원리</div>

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ UI 기획/설계/구현 절차 ○ 유즈케이스에 대한 이해 ○ 데이터 타입 특성 ○ 사용자 기반 메뉴 구조 작성 방법 ○ 감성공학에 대한 이해 【기 술】 ○ 유즈케이스 작성 능력 ○ UI 설계도구 활용 능력 ○ UI 네비게이션 작성 도구 활용 능력 ○ 화면/폼 구성 도구 활용 능력 ○ 화면 구성 능력 ○ UI 작성 그래픽 도구 활용 능력 【태 도】 ○ UI 표준 및 지침을 준수하려는 자세 ○ UI 관련자들과 원활한 관계를 유지하려는 태도 ○ 개발 일정을 준수하려는 태도 ○ UI 설계 변경 요구 피드백에 대해 열린 마음으로 듣는 태도 ○ 고품질의 소프트웨어 개발에 대한 의지

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위를 활용할 경우, 요구사항 도구와 설계 도구의 연계를 고려해야 한다.
- 소프트웨어 아키텍처와 UI표준 및 지침의 변동에 따른 유연성을 확보해야 한다.
- 이 능력단위에 사용되는 기술은 자주 변경되기 때문에 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련이 요구된다.
- 기존의 화면 구현 능력단위의 훈련시간과 수준을 고려하여 화면 설계(5)와 화면 구현(3)으로 분할한다.

자료 및 관련 서류

- 소프트웨어 아키텍처 세부구현 지침
- UI 표준 및 지침
- UI 설계서
- UI 개발기술 매뉴얼
- 한국형 웹 콘텐츠 접근성 지침(KWCAG)
- W3C(World Wide Consortium) 표준
- 전자정부 웹 표준 준수지침

장비 및 도구

- 화면 설계용 컴퓨터
- 대형 모니터
- 프린터, 인터넷
- 네비게이션 설계용 소프트웨어
- UI 설계용 소프트웨어

자료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 화면 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오	√	√
C. 서술형시험		
D. 논술형시험		
E. 사례연구		√
F. 평가자 질문		
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - UI 요건에 대한 이해도
 - UI 설계서에 대한 이해도
 - UI 표준 및 지침에 대한 이해도
 - 최신 UI 기술들에 대한 이해도
 - 사용자 경험 분석에 대한 이해도
 - 프로토타입 작성 및 검증 능력
 - 소프트웨어 아키텍처 상세구현 지침에 대한 이해도
 - UI 설계를 소프트웨어 아키텍처 상세구현 지침과 UI 표준 및 지침에 따라 개발하는 능력

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀워크능력, 리더쉽능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020204_15v3
	보완	2001020224_16v4, 2001020225_16v4
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020225_16v4

능력단위 명칭 : 화면 구현

능력단위 정의 : 화면 구현이란 UI 요구사항을 확인하여 설계한 UI 설계를 기반으로 화면을 구현하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020225_16v4.1 UI 설계 확인하기	<p>1.1 설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다.</p> <p>1.2 UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다.</p> <p>1.3 구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 구현에 요구되는 아키텍처의 이해○ 유용성 개념 및 적용원리○ UI 기획/설계/구현 절차○ 유즈케이스에 대한 이해○ 데이터 타입 특성○ 사용자 기반 메뉴 구조 작성 방법○ 감성공학에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none">○ 유즈케이스 작성 능력○ UI 설계도구 활용 능력○ UI 네비게이션 작성 도구 활용 능력○ 화면/폼 구성 도구 활용 능력○ 화면 구성 능력○ UI 작성 그래픽 도구 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none">○ UI 표준 및 지침을 준수하려는 태도○ UI 관련자들과 원활한 관계를 유지하려는 태도○ 개발 일정을 준수하려는 태도○ UI 설계 변경 요구 피드백에 대해 열린 마음으로 듣는 태도○ 고품질의 소프트웨어 개발에 대한 의지
2001020225_16v4.2 UI 구현하기	<p>2.1 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다.</p> <p>2.2 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다.</p> <p>2.3 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다.</p> <p>2.4 구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 구현에 요구되는 아키텍처의 이해

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 유용성 개념 및 적용원리 ○ HTML 개념과 적용원리 ○ CSS의 개념과 적용원리 ○ 자바스크립트의 개념과 적용원리 ○ 설계 산출물의 이해 ○ 감성공학에 대한 이해 【기 술】 ○ HTML, CSS, 자바스크립트 등 표준기술 활용 능력 ○ UI 네비게이션 작성 도구 활용 능력 ○ 화면/폼 구성 도구 활용 능력 ○ 화면 구성 능력 ○ UI 작성 그래픽 도구 활용 능력 ○ UI Framework(jQuery, AngularJS등) 활용 능력 【태 도】 ○ UI 표준 및 지침을 준수하려는 태도 ○ UI 관련자들과 원활한 관계를 유지하려는 태도 ○ 개발 일정을 준수하려는 태도 ○ UI 설계 변경 요구 피드백에 대해 열린 마음으로 듣는 태도 ○ 고품질의 소프트웨어 개발에 대한 의지

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위를 활용할 경우, 요구사항 도구와 설계 도구의 연계를 고려해야 한다.
- 소프트웨어 아키텍처와 UI표준 및 지침의 변동에 따른 유연성을 확보해야 한다.
- 이 능력단위에 사용되는 기술은 자주 변경되기 때문에 적절한 수행을 위해서는 최신 기술에 대한 지속적인 수집 및 습득을 위한 훈련이 요구된다.
- 기존의 화면 구현 능력단위의 훈련시간과 수준을 고려하여 화면 설계(5)와 화면 구현(3)으로 분할한다.

자료 및 관련 서류

- 소프트웨어 아키텍처 세부구현 지침
- UI 표준 및 지침
- UI 설계서
- UI 개발기술 매뉴얼
- 한국형 웹 콘텐츠 접근성 지침(KWCAG)
- W3C(World Wide Consortium) 표준
- 전자정부 웹 표준 준수지침

장비 및 도구

- 화면 설계용 컴퓨터
- 대형 모니터
- 프린터, 인터넷
- 네비게이션 설계용 소프트웨어
- UI 설계용 소프트웨어

자료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 화면 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오	√	√
C. 서술형시험		
D. 논술형시험		
E. 사례연구		√
F. 평가자 질문		
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - UI 요건에 대한 이해도
 - UI 설계서에 대한 이해도
 - UI 표준 및 지침에 대한 이해도
 - 최신 UI 기술들에 대한 이해도
 - 사용자 경험 분석에 대한 이해도
 - 프로토타입 작성 및 검증 능력
 - 소프트웨어 아키텍처 상세구현 지침에 대한 이해도
 - UI 설계를 소프트웨어 아키텍처 상세구현 지침과 UI 표준 및 지침에 따라 개발하는 능력

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀워크능력, 리더쉽능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020204_15v3
	보완	2001020224_16v4, 2001020225_16v4
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020205_16v4

능력단위 명칭 : 데이터 입출력 구현

능력단위 정의 : 데이터 입출력 구현이란 응용소프트웨어가 다루어야하는 데이터 및 이들 간의 연관성, 제약조건을 식별하여 논리적으로 조직화 하고, 소프트웨어 아키텍처에 기술된 데이터 저장소에 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	<p>1.1 업무 분석가, 데이터베이스 엔지니어가 작성한 논리 데이터저장소 설계 내역에서 정의된 데이터의 유형을 확인하고 식별할 수 있다.</p> <p>1.2 논리 데이터저장소 설계 내역에서 데이터의 논리적 단위와 데이터 간의 관계를 확인할 수 있다.</p> <p>1.3 논리 데이터저장소 설계 내역에서 데이터 또는 데이터간의 제약조건과 이들 간의 관계를 식별할 수 있다.</p> <p>【지 식】</p> <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) <p>【기 술】</p> <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 <p>【태 도】</p> <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
2001020205_16v4.2 물리 데이터저장소 설계하기	<p>2.1 논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을 반영한 물리 데이터저장소 설계를 수행할 수 있다.</p> <p>2.2 논리 데이터저장소 설계를 바탕으로 목표 시스템의 데이터 특성을 반영하여 최적화된 물리 데이터저장소를 설계할 수 있다.</p> <p>2.3 물리 데이터저장소 설계에 따라 데이터저장소에 실제 데이터가 저장될 물리적 공간을 구성할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <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) <p>【기 술】</p> <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 <p>【태 도】</p> <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
<p>2001020205_16v4.3</p> <p>데이터 조작 프로시저 작성하기</p>	<p>3.1 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소에 연결을 수행하는 프로시저를 작성할 수 있다.</p> <p>3.2 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소로부터 데이터를 읽어 오는 프로시저를 작성할 수 있다.</p> <p>3.3 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 변경 내용 또는 신규 입력된 데이터를 데이터 저장소에 저장하는 프로시저를 작성할 수 있다.</p> <p>3.4 구현된 데이터 조작 프로시저를 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.</p> <p>【지 식】</p> <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 ○ 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등) ○ SQL(Structure Query Language) ○ 소프트웨어 테스트에 대한 이해 <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 ○ ORM 프레임워크 활용 능력 ○ 단위테스트 도구 활용기법 ○ 쿼리(Query) 성능 측정 도구 활용 능력 <p>【태 도】</p>

능 력 단 위 요 소	수 행 준 거
	<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	<p>4.1 프로그래밍 언어와 도구에 대한 이해를 바탕으로 응용소프트웨어 설계, 물리 데이터저장소 설계와 운영 환경을 고려하여 데이터 조작 프로시저의 성능을 예측할 수 있다.</p> <p>4.2 업무 분석가에 의해 정의된 요구사항을 기준으로, 성능측정 도구를 활용하여 데이터 조작 프로시저의 성능을 측정할 수 있다.</p> <p>4.3 실 데이터를 기반으로 테스트를 수행하여 데이터 조작 프로시저의 성능에 영향을 주는 병목을 파악할 수 있다.</p> <p>4.4 테스트 결과와 정의된 요구사항을 기준으로 데이터조작 프로시저의 성능에 따른 이슈 발생 시 이에 대해 해결할 수 있다.</p> <p>【지 식】</p> <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 <p>【기 술】</p> <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 <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

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 응용 소프트웨어 개발을 위해 필요한 애플리케이션 개발 업무에 적용된다.
- 이 능력단위는 논리데이터저장소 설계 시 도출된 산출물을 충실히 반영하고, 개발 소스코드 작성을 위한 지침을 기반으로 수행하여야 한다.
- 이 능력단위의 적절한 수행을 위해서는 데이터저장소 설계 기술과 다양한 데이터저장소 제품의 특성 및 관련 표준에 대한 지속적인 훈련이 필요하다.

자료 및 관련 서류

- UML 작성표준
- ERD 작성표준
- 개발 시 활용 데이터 저장소에 대한 매뉴얼 및 개발 가이드
- 아키텍처 정의서
- 모델링 검토기준

장비 및 도구

- 서버, 데스크탑, 노트북, 프린터
- 네트워크, 인터넷
- 서버프로그램,
- 형상관리 프로그램
- 데이터저장소 설계 프로그램
- 데이터 입출력 성능 측정 프로그램

자료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	
B. 문제해결 시나리오	√	
C. 서술형시험		√
D. 논술형시험		√
E. 사례연구		
F. 평가자 질문	√	
G. 평가자 체크리스트		√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 업무영역에서 발생하는 데이터와 그 용처에 대한 이해
 - 분석 산출물 및 다이어그램 분석
 - 데이터 저장소 관련 설계 산출물 및 다이어그램 이해
 - 개발 애플리케이션의 주요 매커니즘 및 적용 개발언어 분석 능력
 - 데이터 조작 전용 프로그래밍 언어의 이해
 - 이름규칙, 주석처리 등의 이해
 - 형상관리 도구 방법과 소프트웨어 아키텍처와 일관성
 - 사용하는 물리데이터저장소의 성능과 특성에 대해 이해

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	수리능력	기초연산능력, 기초통계능력, 도표분석능력, 도표작성능력
3	자원관리능력	시간자원관리능력, 예산자원관리능력, 물적자원관리능력, 인적자원관리능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020205_15v3
	보완	2001020201_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

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

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 단위테스트 기술동향 및 도구 ○ 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	<p>4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.</p> <p>4.2 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.</p> <p>4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.</p> 【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

능 력 단 위 요 소	수 행 준 거
	○ 적극적인 업무 분석 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 응용소프트웨어 개발을 위해 필요한 애플리케이션 개발 업무에 적용된다.
- 이 능력단위는 애플리케이션 설계 시 도출된 산출물을 충실히 반영하고, 개발 소스코드 작성을 위한 지침을 기반으로 수행하여야 한다.
- 이 능력단위의 적절한 수행을 위해서는 애플리케이션 구현 기술에 대한 지속적인 훈련이 필요하다.
- 이 능력단위에서 언급하는 배치 프로그램은 서버에서 일괄 처리되는 모듈을 의미한다.

자료 및 관련 서류

- IEEE 표준 1074 - 소프트웨어 개발 필요한 절차
- IEEE 표준 1008 - 소프트웨어 단위 테스트 기법
- TTAS.KO-11.0030 : 소프트웨어 컴포넌트 개발 산출물 명세 표준
- TTAS.IE-1008 : 소프트웨어 단위 시험 표준
- 개발 시 활용 컴포넌트 사용에 대한 매뉴얼 및 개발 가이드
- 개발 시 적용되는 개발표준 문서
- 테스트에 대한 계획/설계/결과보고/결함에 대한 원인분석보고 문서

장비 및 도구

- 서버, 데스크탑, 노트북, 프린터
- 네트워크, 인터넷
- 형상관리 프로그램
- 개발전용 프로그램

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오	√	√
C. 서술형시험		
D. 논술형시험		
E. 사례연구		
F. 평가자 질문		
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	√
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 분석/설계 산출물 및 다이어그램 분석
 - 개발 애플리케이션의 주요 매커니즘 및 적용 개발언어 분석 능력
 - 코드 이름규칙, 주석처리 방법
 - 형상관리 절차 및 기준의 이해
 - 서버프로그램 공통 리소스(예:데이터베이스) 활용의 모듈화 이해
 - 프로그램 언어별 데이터베이스 연동 방법 이해
 - 프로그램 언어별 웹 서버프로그램 개발 방법의 이해

- 서버프로그램에서 적용하는 주요 모듈구성방법(레이어별 패턴)의 이해
- 컴포넌트 코드작성의 이해
- 프로토콜에 따른 클라이언트와 서버 프로그램의 요청 및 응답 매커니즘 이해
- 테스트를 위한 유효값 입력, 비유효값 입력, 예외 조건의 이해
- 배치프로그램을 위한 데이터베이스와 연관된 메모리 활용방법의 이해
- 네트워크 및 파일처리 기반의 배치프로그램 작성방법 이해
- 배치프로그램 트랜잭션 처리 방법의 이해

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020211_15v3
	보완	2001020211_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020212_16v4

능력단위 명칭 : 인터페이스 구현

능력단위 정의 : 인터페이스 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020212_16v4.1 인터페이스 설계서 확인하기	<p>1.1 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다.</p> <p>1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다.</p> <p>1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 산업 분야별 업무특성○ 개발 방법론○ 소프트웨어 아키텍처○ 설계 모델링 기법○ 디자인 패턴○ 네트워크○ 분산 프로그래밍 <p>【기 술】</p> <ul style="list-style-type: none">○ UML 작성 기술○ 설계 모델링 기술○ IDE 및 개발환경 도구 활용○ 프레임워크 활용○ 기술영역별 미들웨어/솔루션 활용○ 데이터 연계/이관 도구 활용 능력○ 데이터 보안 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 고객의 요청을 적극적으로 수용하고자 하는 태도○ 정확성과 완전성을 기하고자 하는 의지○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도○ 산출물 완성도를 위한 적극적인 태도○ 주어진 과제를 완수하는 책임감
2001020212_16v4.2 인터페이스 기능 구현하기	<p>2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다.</p> <p>2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>2.3 개발하고자하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다.</p> <p>2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 산업 분야별 업무특성 ○ 개발 방법론 ○ 소프트웨어 아키텍처 ○ 설계 모델링 기법 ○ 디자인 패턴 ○ 네트워크 ○ 분산 프로그래밍 <p>【기 술】</p> <ul style="list-style-type: none"> ○ UML 작성 기술 ○ 설계 모델링 기술 ○ IDE 및 개발환경 도구 활용 ○ 프로그램 언어 활용 능력 ○ 프레임워크 활용 능력 ○ 기술영역별 미들웨어/솔루션 활용 기술 ○ 데이터 연계/이관 도구 활용 능력 ○ 데이터 보안 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 정확성과 완전성을 기하고자 하는 의지 ○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 ○ 산출물 완성도를 위한 적극적인 태도 ○ 주어진 과제를 완수하는 책임감
<p>2001020212_16v4.3</p> <p>인터페이스 구현 검증하기</p>	<p>3.1 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다.</p> <p>3.2 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다.</p> <p>3.3 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 설계 산출물 ○ 디자인 패턴 ○ 데이터베이스 ○ 네트워크 ○ 분산 프로그래밍 ○ 소프트웨어 테스트 <p>【기 술】</p>

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ IDE 도구 활용 능력 ○ 프로그램 언어 활용 능력 ○ 프로그램 디버깅 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 데이터 연계/이관 도구 활용 능력 ○ 데이터 보안 기술 ○ 프로그램 코드검토 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

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 시스템 소프트웨어, 미들웨어 등의 소프트웨어 인프라에 의존적인 측면이 있으므로 특정 공급자의 제품 특성을 반영하지 않도록 적용한다.
- 이 능력단위에서 사용되는 기술은 변화 속도가 빠르고 새로운 방식의 접근이 필요한 경우가 많기 때문에 새로운 기술에 대한 지속적인 습득과 학습 등을 적용한다.
- 이 능력단위는 업무 환경, 시스템 환경, 기능적/비 기능적 요구사항 등에 따라 서로 다른 솔루션을 필요로 하므로 주어진 조건에 맞는 해결방안을 수립하기 위해 유사 사례에 대한 분석과 베스트 프랙티스의 활용 등을 포함한다.

자료 및 관련 서류

- 대내/외 인터페이스 표준 및 명세서
- 연계 데이터 표준 및 설계서
- 표준 및 절차 매뉴얼
- W3C 웹서비스 표준 아키텍처 (W3C 아키텍처 실무그룹)

장비 및 도구

- 서버, 데스크탑, 노트북
- 네트워크, 인터넷, 데이터베이스, 저장장치
- 인터페이스 미들웨어 (EAI/MCI, ESB)
- 시스템 모니터링 소프트웨어/도구
- 네트워크 모니터링 소프트웨어/도구
- 트랜잭션 모니터링 소프트웨어/도구
- 성능 모니터링 소프트웨어/도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 통합 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오		√
C. 서술형시험		√
D. 논술형시험		√
E. 사례연구	√	√
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	
I. 일지/저널		
J. 역할연기		
K. 구두발표		√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 연계 매카니즘에 대한 이해도
 - 연계 표준/프로토콜에 대한 이해도
 - 미들웨어, 인터페이스 소프트웨어에 대한 이해도
 - 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020212_15v3
	보완	2001020212_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019-2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020206_16v4

능력단위 명칭 : 통합 구현

능력단위 정의 : 통합 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하여 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020206_16v4.1 연계 데이터 구성하기	<div>1.1 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다.</div> <div>1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다.</div> <div>1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.</div> <div>【지 식】</div> <div>○ 해당 산업 분야에 대한 지식</div> <div>○ 업무 특성에 대한 이해</div> <div>○ 개발 방법론 이해</div> <div>○ 소프트웨어 아키텍처에 대한 이해</div> <div>○ 설계 모델링 기법</div> <div>○ 네트워크 이해</div> <div>○ 분산 프로그래밍에 대한 이해</div> <div>【기 술】</div> <div>○ UML 작성 기술</div> <div>○ 설계 모델링 기술</div> <div>○ IDE 및 개발환경 도구 활용</div> <div>○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용</div> <div>○ 기술영역별 미들웨어/솔루션 활용</div> <div>○ 데이터 연계/이관 도구 활용 능력</div> <div>○ 데이터 보안 기술</div> <div>【태 도】</div> <div>○ 고객의 요청을 적극적으로 수용하고자 하는 태도</div> <div>○ 정확성과 완전성을 기하고자 하는 의지</div> <div>○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도</div> <div>○ 산출물 완성도를 위한 적극적인 태도</div> <div>○ 주어진 과제를 완수하는 책임감</div>
2001020206_16v4.2 연계 매커니즘 구성하기	<div>2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다.</div> <div>2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다.</div> <div>2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다.</div> <div>2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시</div>

능 력 단 위 요 소	수 행 준 거
	<p>보안을 적용할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 개발 방법론 이해 ○ 소프트웨어 아키텍처에 대한 이해 ○ 설계 모델링 기법 ○ 네트워크 이해 ○ 분산 프로그래밍에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ UML 작성 기술 ○ 설계 모델링 기술 ○ IDE 및 개발환경 도구 활용 ○ 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 ○ 기술영역별 미들웨어/솔루션 활용 ○ 데이터 연계/이관 도구 활용 능력 ○ 데이터 보안 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객의 요청을 적극적으로 수용하고자 하는 태도 ○ 정확성과 완전성을 기하고자 하는 의지 ○ 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 ○ 산출물 완성도를 위한 적극적인 태도 ○ 주어진 과제를 완수하는 책임감
<p>2001020206_16v4.3</p> <p>내외부 연계 모듈 구현하기</p>	<p>3.1 구성된 연계 메카니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다.</p> <p>3.2 구성된 연계 메카니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다.</p> <p>3.3 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다.</p> <p>3.4 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처의 이해 ○ 설계 산출물의 이해 ○ 프로그래밍언어 활용방법의 이해 ○ 데이터베이스 이해 ○ 네트워크 이해 ○ 분산 프로그래밍에 대한 이해 ○ 소프트웨어 테스트에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ IDE 도구 활용 능력 ○ 프로그램 디버깅 기법 ○ 형상관리 도구 활용 기법 ○ 단위테스트 도구 활용기법 ○ 데이터 연계/이관 도구 활용 능력

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 데이터 보안 기술 【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

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 시스템 소프트웨어, 미들웨어 등의 소프트웨어 인프라에 의존적인 측면이 있으므로 특정 공급자의 제품 특성을 반영하지 않도록 해야 한다.
- 이 능력단위에서 사용되는 기술은 변화 속도가 빠르고 새로운 방식의 접근이 필요한 경우가 많기 때문에 새로운 기술에 대한 지속적인 습득과 학습이 필요하다.
- 이 능력단위는 업무 환경, 시스템 환경, 기능적/비 기능적 요구사항 등에 따라 서로 다른 솔루션을 필요로 하므로 주어진 조건에 맞는 해결방안을 수립하기 위해 유사 사례에 대한 분석과 베스트 프랙티스의 활용이 필요하다.

자료 및 관련 서류

- 대내/외 인터페이스 표준 및 명세서
- 연계 데이터 표준 및 설계서
- 표준 및 절차 매뉴얼
- W3C 웹서비스 표준 아키텍처 (W3C 아키텍처 실무그룹)

장비 및 도구

- 서버, 데스크탑, 노트북
- 네트워크, 인터넷, 데이터베이스, 저장장치
- 인터페이스 미들웨어 (EAI/MCI, ESB)
- 시스템 모니터링 소프트웨어/도구
- 네트워크 모니터링 소프트웨어/도구
- 트랜잭션 모니터링 소프트웨어/도구
- 성능 모니터링 소프트웨어/도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 통합 구현의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오		√
C. 서술형시험		√
D. 논술형시험		
E. 사례연구	√	√
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	
I. 일지/저널		
J. 역할연기		
K. 구두발표		√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 연계 매카니즘에 대한 이해도
 - 연계 표준/프로토콜에 대한 이해도
 - 미들웨어, 인터페이스 소프트웨어에 대한 이해도
 - 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	수리능력	기초연산능력, 기초통계능력, 도표분석능력, 도표작성능력
3	대인관계능력	팀워크능력, 리더쉽능력, 갈등관리능력, 협상능력, 고객서비스능력
4	기술능력	기술이해능력, 기술선택능력, 기술적용능력
5	조직이해능력	국제감각, 조직체제이해능력, 경영이해능력, 업무이해능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020206_15v3
	보완	2001020206_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

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

● 적용범위 및 작업상황

고려사항

- 이 능력단위는 정보서비스 개발을 위한 개발단계별 산출물에 대한 동료검토(Peer Review) 활동을 제외한 제반 테스트 활동을 포함한다.
- 비기능적 요구사항인 효율성, 신뢰성, 이식성, 성능, 보안, 상호운용성, 안정성 등과 같은 기능외적인 요구사항을 고려해야 한다.
- 기 수립한 테스트계획에 따라 테스트 케이스를 작성하고 테스트를 수행하여야 한다.
- 이 능력단위는 수준을 고려하여 기존의 애플리케이션 테스트를 애플리케이션 테스트 관리(5)와 애플리케이션 테스트 수행(3)으로 분할한다.

자료 및 관련 서류

- 국제 표준 규격(ISO/IEC 29119, ISO/IEC 25000, ISO/IEC 9126, IEEE 829) 기준
- ISTQB Syllabus(테스트 지식체계)
- CSTE CBOK(테스트 지식체계)

장비 및 도구

- 컴퓨터
- 대형 모니터
- 프린터, 인터넷

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 테스트 수행의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트	√	
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 테스트 계획의 구성항목을 이해하고 테스트 케이스 작성 능력
 - 테스트 전략을 이해하고 테스트 케이스 작성 시 적용할 수 있는 능력
 - 테스트 케이스를 이해하고 테스트를 수행할 수 있는 능력
 - 테스트 관련 국제 표준 지식체계에 대하여 이해하고 숙지하고 있는지의 여부

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	수리능력	기초연산능력, 기초통계능력, 도표분석능력, 도표작성능력
3	문제해결능력	사고력, 문제처리능력
4	자원관리능력	시간자원관리능력, 예산자원관리능력, 물적자원관리능력, 인적자원관리능력
5	대인관계능력	팀워크능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
6	정보능력	컴퓨터활용능력, 정보처리능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020213_15v3
	보완	2001020226_16v4, 2001020227_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020226_16v4

능력단위 명칭 : 애플리케이션 테스트 관리

능력단위 정의 : 애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020226_16v4.1 애플리케이션 테스트 케이스 설계하기	<p>1.1 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다.</p> <p>1.2 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다.</p> <p>1.3 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비 할 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 자세○ 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
2001020226_16v4.2 애플리케이션 개발자 통합 테스트하기	<p>2.1 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.</p> <p>2.2 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정 할 수 있다.</p> <p>2.3 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 소프트웨어 아키텍처에 대한 이해 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)에 대한 지식 ○ 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법 ○ 결함 관리에 대한 지식 <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
<p>2001020226_16v4.3</p> <p>애플리케이션 성능 개선하기</p>	<p>3.1 애플리케이션 테스트를 통하여 애플리케이션의 성능을 분석하고, 성능 저하 요인을 발견할 수 있다.</p> <p>3.2 코드 최적화 기법, 아키텍처 조정 및 호출 순서 조정 등을 적용하여 애플리케이션 성능을 개선할 수 있다.</p> <p>3.3 프로그래밍 언어의 특성에 대한 이해를 기반으로 소스코드 품질 분석 도구를 활용하여 애플리케이션 성능을 개선할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 구현에 요구되는 아키텍처 ○ 프로그램 언어 활용 방법 ○ 소스코드 인스펙션 ○ 프로그래밍 표준 ○ 성능 평가 기준 ○ 알고리즘 ○ 자료구조 <p>【기 술】</p> <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 ○ 소스코드 품질분석 도구(PMD, SONAR 등) <p>【태 도】</p>

능 력 단 위 요 소	수 행 준 거
	<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

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 본격적인 테스트(단위, 통합, 시스템, 인수 테스트)를 수행하기 전 애플리케이션 개발자 입장에서 자신이 개발한 프로그램에 대한 디버깅 및 간단한 테스트를 포함한다.
- 애플리케이션 테스트는 비 기능적 요구사항인 효율성, 신뢰성, 이식성, 성능, 보안, 상호운용성, 안정성 등과 같은 기능외적인 요구사항도 고려해야 한다.
- 본격적인 테스트(단위, 통합, 시스템, 인수 테스트)는 IT테스트 세분류를 참조한다.
- 이 능력단위는 수준을 고려하여 기존의 애플리케이션 테스트를 애플리케이션 테스트 관리(5)와 애플리케이션 테스트 수행(3)으로 분할한다.

자료 및 관련 서류

- 국제 표준 규격(ISO/IEC 29119, ISO/IEC 25000, ISO/IEC 9126, IEEE 829) 기준
- ISTQB Syllabus(테스트 지식체계)
- CSTE CBOK(테스트 지식체계)

장비 및 도구

- 컴퓨터
- 대형 모니터
- 프린터, 인터넷

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 개발자 테스트의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험		
D. 논술형시험		
E. 사례연구	√	√
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		
I. 일지/저널	√	√
J. 역할연기		
K. 구두발표		
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 애플리케이션 성능 개선 평가 시 문제해결 시나리오를 활용하여 문제 해결 여부를 확인한다.
 - 사례연구는 개발자 테스트 수행 사례를 도출하여 점검한다.
 - 평가자 질문은 개발자 테스터에게 질문을 통하여 능력 향상여부를 확인한다.
 - 평가자 체크리스트는 개발자 테스트 수행 결과를 평가자가 직접 점검하여 체크리스트에 반영한다.
 - 피평가자 체크리스트는 개발자 테스터가 직접 자가 점검할 때에 사용한다.

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀워크능력, 리더쉽능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020213_15v3
	보완	2001020226_16v4, 2001020227_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

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

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 응용SW엔지니어링 애플리케이션 배포 업무에 적용한다.
- 애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포하는 것을 말한다.
- 이 능력단위는 응용SW 애플리케이션 아키텍처와 개발 환경에 대한 이해를 바탕으로 빌드와 배포 환경을 구성해야 한다.
- 응용 SW 개발 환경의 형상관리 도구와 빌드/배포 도구 등에 대한 이해를 바탕으로 자동화 도구를 활용하여 애플리케이션 배포 업무를 수행할 수 있어야 한다.
- 응용 SW의 개발 언어, 환경, 플랫폼, 아키텍처에 따라 빌드 및 배포 절차와 방법이 달라 질 수 있으므로 다양한 환경 적용할 수 있도록 지속적인 학습이 필요하다.

자료 및 관련 서류

- 소스코드 검증 지침
- 소스코드 검증 도구 매뉴얼
- 애플리케이션 형상관리 지침서
- 애플리케이션 형상관리 도구 매뉴얼
- 애플리케이션 빌드 지침서
- 애플리케이션 빌드 절차 설명서
- 애플리케이션 빌드 도구 매뉴얼
- 애플리케이션 배포 지침서
- 애플리케이션 운영 배포 절차 설명서
- 애플리케이션 배포 도구 매뉴얼

장비 및 도구

- 애플리케이션 배포 환경 : 형상관리 도구, 빌드 도구, 배포 도구 등
- 애플리케이션 검증 도구 : 소스코드 검증 도구, 정적 소스코드 분석 도구, 동적 소스코드 분석 도구 등
- 전산장비 : 인터넷, 컴퓨터, 프린터, 복사기, 빔 프로젝터, 등
- 지원도구 : 문서작성 도구 등
- OA소프트웨어 : 워드프로세서, 스프레드시트, 프레젠테이션 도구 등

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 애플리케이션 배포의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오 (윤고딕110, 10p, 90%, -10)		
B. 문제해결 시나리오	√	
C. 서술형시험		√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문		√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		
I. 일지/저널	√	
J. 역할연기		
K. 구두발표	√	
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 애플리케이션 개발 및 운영 환경에 대한 이해 여부 점검

- 애플리케이션 배포 환경 구성을 위한 관련 도구 설치 및 설정 가능 여부 평가
- 빌드 환경 구성에 따른 각종 도구 사용 능력 평가
- 소스코드 검증 기법에 대한 이해를 바탕으로 도구 사용 능력 점검
- 애플리케이션 빌드 및 배포 절차에 대한 이해 및 도구 사용 능력 확인
- 애플리케이션 빌드 및 배포 스크립트 작성 실습 및 점검
- 애플리케이션 빌드 및 배포 절차에 따라 빌드 실행 능력 점검
- 애플리케이션 빌드 결과에 대한 분석 및 이해 능력 점검
- 애플리케이션 빌드 실패 시 원복 수행 절차 이해 및 실시 여부 확인

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020214_15v3
	보완	2001020214_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020208_16v3

능력단위 명칭 : 정보시스템 이행

능력단위 정의 : 정보시스템 이행이란 개발자 환경에서 개발한 결과물을 운영 환경에 설치하고, 사용자 요구사항과 최종적으로 일치하는 지에 대해 승인을 얻어 응용소프트웨어 결과물을 사용자에게 전달하여 인계하고 시스템을 운영할 수 있도록 교육하고 지원하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020208_16v3.1 정보시스템 사용자 교육하기	<p>1.1 개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다.</p> <p>1.2 개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다.</p> <p>1.3 효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검할 수 있다.</p> <p>1.4 교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육결과 및 성과를 평가할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 해당 산업 분야에 대한 지식○ 업무 특성에 대한 이해○ 소프트웨어 아키텍처에 대한 이해○ 문서작성 도구 활용 방법○ 표준 양식(Template) 작성 및 활용법○ 국제 표준 제품 품질특성에 대한 사항○ 피교육 집단 특성에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none">○ 문서작성 도구 활용 능력○ 프리젠테이션 기법○ 교육성과 측정 기법○ 교육성과 분석 기법○ 멀티미디어 작성 도구 활용 능력○ 표준 용어집 <p>【태 도】</p> <ul style="list-style-type: none">○ 교재의 일관성 확보를 위해 노력하는 태도○ 교육효과를 높이기 위한 기자재 및 장소 확보 의지○ 표준 양식 준수 의지○ 사용자 질의사항에 대한 긍정적 태도○ 다양한 이해당사자의 의견을 긍정적으로 수용할 수 있는 태도
2001020208_16v3.2 정보시스템 이행하기	<p>2.1 하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다.</p> <p>2.2 시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>2.3 정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증할 수 있다.</p> <p>2.4 정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 소프트웨어 아키텍처에 대한 이해 ○ 시스템 운영 환경에 대한 이해 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 시스템 운영에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 ○ 시스템 성능 분석 및 진단에 대한 지식 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 애플리케이션 패키징 기술 ○ 애플리케이션 배포 도구 활용 능력 ○ 애플리케이션 모니터링 도구 활용 능력 ○ 형상관리 도구 활용 기법 ○ 결함관리 도구 활용 능력 ○ 성능 측정 도구 활용 능력 <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 ○ 성공적 시스템 이행을 위한 의지
2001020208_16v3.3 정보시스템 안정화하기	<p>3.1 정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다.</p> <p>3.2 정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정보시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다.</p> <p>3.3 개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.</p> <hr/> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 해당 산업 분야에 대한 지식 ○ 업무 특성에 대한 이해 ○ 소프트웨어 아키텍처에 대한 이해 ○ 시스템 운영 환경에 대한 이해 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 시스템 운영에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 ○ 시스템 성능 분석 및 진단에 대한 지식 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 애플리케이션 패키징 기술 ○ 애플리케이션 배포 도구 활용 능력 ○ 애플리케이션 모니터링 도구 활용 능력 ○ 형상관리 도구 활용 기법 ○ 결함관리 도구 활용 능력

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 성능 측정 도구 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 태도 ○ 기술 및 제품 동향에 관한 각종 정보 수집에 대한 적극성 ○ 오류 및 추가 요구사항에 대한 반영 의지 ○ 시스템 성능의 문제점 및 성능 개선을 위한 의지 ○ 시스템 안정화를 위한 의지

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 개발자 환경에서 개발된 결과물을 실제 시스템이 운영될 사용자 환경에 설치하여, 개발된 시스템으로 전환함으로써 원활한 운영이 가능하도록 하는 업무에 적용한다.
- 사용자 교육 수행 시 사용자의 눈높이를 고려하여 교육을 수행한다.

자료 및 관련 서류

- 국내외 규격(KS, JIS, ASTM, SAE 등)
- ISO 기준
- 원가계산서, 개발계획서, 견적서
- 개발검토서
- 작업표준서, 검사성적서, 검사기준서
- 작업일지
- 설비점검표

장비 및 도구

- 이행상황 점검 장비: 컴퓨터, 인터넷, 프린터
- 이행 장비: 시스템, 네트워크, 데이터베이스

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 정보시스템 이행의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 교육일정 계획의 실현가능성과, 교육시행 후 교육성과
 - 작성된 매뉴얼(사용자, 개발자, 운영자 등)의 구체성과 이해 용이성
 - 시스템 운영환경 설정과 애플리케이션의 배치 상황
 - 이해 시간대별 영역별 사전점검표의 구체성
 - 데이터 이행작업 점검 및 검증
 - 이행여부 결정을 위한 과정 및 결정근거
 - 이행 후 오류 및 조치 내역과 사용자 추가요구사항 수집 및 반영내역
 - 시스템 안정화 및 성능개선을 위한 조치 내역

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자원관리능력	시간자원관리능력, 예산자원관리능력, 물적자원관리능력, 인적자원관리능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	조직이해능력	국제감각, 조직체제이해능력, 경영이해능력, 업무이해능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020208_14v2
	보완	2001020208_16v3
개발연도	현재	2016
	2차	2014
	최초(1차)	-
버전번호		v3
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020217_16v4

능력단위 명칭 : 애플리케이션 리팩토링

능력단위 정의 : 애플리케이션 리팩토링이란 소스코드가 수행하는 기능을 유지하면서 코드 가독성 및 잠재적 결함을 제거하도록 코드의 구조를 개선하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020217_16v4.1 리팩토링 기준 수립하기	<p>1.1 리팩토링을 적용하여 달성하려는 품질 목표를 수립할 수 있다.</p> <p>1.2 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다.</p> <p>1.3 리팩토링에 활용할 수 있는 도구 및 절차를 수립할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 비기능적 품질 특성○ 클린 코드의 필요성 및 특징○ 국제 코딩 표준○ 리팩토링 기법○ 디자인 패턴○ 회귀 테스트 <p>【기 술】</p> <ul style="list-style-type: none">○ 코드 정적 분석 도구 활용 기술○ 코드 동적 분석 도구 활용 기술○ 역공학 도구 활용 기술○ 체크리스트 작성 기술○ 테스트 도구 활용 기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 이해하기 쉽고 확장하기 쉬운 코드로 개선하려는 태도○ 결함 없는 코드로 개선하려는 태도○ 지속적으로 리팩토링을 수행하려는 적극적 태도
2001020217_16v4.2 소스 코드 분석하기	<p>2.1 개발 표준을 만족하지 않는 소스 코드를 식별할 수 있다.</p> <p>2.2 성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별할 수 있다.</p> <p>2.3 잠재 결함 요소를 내포한 소드 코드를 식별할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 비기능적 품질 특성 충족 기법○ 국제 코딩 표준○ 코드 스멜 유형○ 리팩토링 기법○ 디자인 패턴 <p>【기 술】</p> <ul style="list-style-type: none">○ 코드 인스펙션 기술○ 설계 패턴 적용 기술○ 코드 정적 분석 도구 활용 기술

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 코드 동적 분석 도구 활용 기술 ○ 역공학 도구 활용 기술 ○ 리팩토링 도구 활용 능력 ○ IDE 도구 활용 능력 ○ 좋은 설계 (Good Design) 기준 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 이해하기 쉽고 확장하기 쉬운 코드로 개선하려는 태도 ○ 결함 없는 코드로 개선하려는 태도 ○ 지속적으로 리팩토링을 수행하려는 적극적 태도
<p>2001020217_16v4.3</p> <p>리팩토링 적용하기</p>	<p>3.1 리팩토링을 위한 소스 코드 구조를 개선할 수 있다.</p> <p>3.2 개선한 소스 코드가 초기 기능을 유지하는지 확인할 수 있다.</p> <p>3.3 개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 비기능적 품질 특성 ○ 클린 코드 특징 ○ 국제 코딩 표준 ○ 리팩토링 적용 기법 ○ 디자인 패턴 ○ 회귀 테스트 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 설계 패턴 적용 기술 ○ 리팩토링 도구 활용 능력 ○ 코드 정적 분석 도구 활용 기술 ○ 코드 동적 분석 도구 활용 기술 ○ IDE 도구 활용 능력 ○ 코드 인스펙션 기술 ○ 프로그램 디버깅 능력 ○ 테스트 도구 활용 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 이해하기 쉽고 확장하기 쉬운 코드로 개선하려는 태도 ○ 결함 없는 코드로 개선하려는 태도 ○ 지속적으로 리팩토링을 수행하려는 적극적 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 도구를 활용하여 코드 표준 준수 여부를 파악 할 수 있어야 한다.
- 이 능력단위는 코드 리팩토링 후 원래 달성해야 하는 기능 목적을 준수하는지 확인할 수 있는 회귀테스트 환경을 갖추고 있어야 한다.
- 이 능력단위는 비기능적 요구사항을 만족하기 위한 활동으로 품질 목표 달성을 확인할 수 있는 체크리스트를 확보해야 한다.
- 이 능력단위는 리팩토링 기법으로서의 클린 코드, 설계 패턴에 대한 이해 및 패턴 적용을 통한 코드 개선 기법을 파악하고 있어야 한다.

자료 및 관련 서류

- 코드 표준 및 절차 매뉴얼
- 분석 및 설계 패턴
- 코드 품질 체크 리스트

장비 및 도구

- 컴퓨터, 인터넷
- 통합 개발 환경
- 코드 분석 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 리팩토링 적용하기의 수행준거에 제시되어 있는 내용을 평가하기 위해 문제 해결 중심의 사례 연구와 과정과 결과를 평가할 수 있는 포트폴리오 기반의 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오	√	√
B. 문제해결 시나리오		
C. 서술형시험		
D. 논술형시험		
E. 사례연구	√	√
F. 평가자 질문	√	√
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기		
K. 구두발표		
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 리팩토링 전/후의 기능성 유지 여부 확인
 - 분석 대상에서 발견한 개선 점의 정확성 여부 확인
 - 적용한 패턴의 정확성 확인

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020217_15v3
	보완	2001020217_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019-2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020209_16v4

능력단위 명칭 : 제품소프트웨어 패키징

능력단위 정의 : 제품소프트웨어 패키징이란 개발이 완료된 제품소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전 관리를 수행하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020209_16v4.1 제품소프트웨어 패키징하기	<p>1.1 신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다.</p> <p>1.2 이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다.</p> <p>1.3 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 제품소프트웨어 관련 시장에 대한 지식○ 제품소프트웨어가 적용되는 업무 특성에 대한 이해○ 제품소프트웨어 아키텍처에 대한 이해○ 사용자의 제품 실행 환경에 대한 이해○ 국제 표준 제품 품질특성에 대한 사항○ 제품소프트웨어 사용에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해○ 제품 패키징, 배포 관련 표준에 대한 지식 <p>【기 술】</p> <ul style="list-style-type: none">○ 애플리케이션 패키징 기술○ 애플리케이션 배포 도구 활용 능력○ 애플리케이션 모니터링 도구 활용 능력○ 형상관리 도구 활용 기법○ 결함관리 도구 활용 능력○ 암호화/보안 모듈 (DRM) 탑재 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○ 제품소프트웨어의 특성에 대해 이해하고자 하는 태도○ 사용자 중심으로 제품소프트웨어 활용에 대해 이해하고자 하는 태도
2001020209_16v4.2 제품소프트웨어 매뉴얼 작성하기	<p>2.1 사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.</p> <p>2.2 사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.</p> <p>2.3 사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ 제품소프트웨어 관련 시장에 대한 지식 ○ 제품소프트웨어가 적용되는 업무 특성에 대한 이해 ○ 제품소프트웨어 아키텍처에 대한 이해 ○ 사용자의 제품 실행 환경에 대한 이해 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 릴리즈 노트 작성법 ○ 각종 매뉴얼 작성법 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 문서작성 도구 활용 능력 ○ 프리젠테이션 기법 ○ 도움말 생성 기술 ○ API 문서 생성 기술(JavaDoc, doxygen) ○ 멀티미디어 작성 도구 활용 능력 ○ 표준 용어집 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 교재의 일관성 확보를 위해 노력하는 태도 ○ 표준 양식 준수 의지 ○ 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도 ○ 제품소프트웨어의 특성에 대해 이해하고자 하는 태도 ○ 사용자 중심으로 제품소프트웨어 활용에 대해 이해하고자 하는 태도
2001020209_16v4.3 제품소프트웨어 버전관리하 기	<p>3.1 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등 록할 수 있다.</p> <p>3.2 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다.</p> <p>3.3 버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 제품소프트웨어 관련 시장에 대한 지식 ○ 제품소프트웨어가 적용되는 업무 특성에 대한 이해 ○ 제품소프트웨어 아키텍처에 대한 이해 ○ 사용자의 제품 실행 환경에 대한 이해 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 장애 복구 처리 절차 ○ 소프트웨어 버전관리 및 도구 사용에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 협업도구 활용 능력 ○ 형상관리 도구 활용 기법 ○ 결함관리 도구 활용 능력 ○ 성능 측정 도구 활용 능력 ○ 소프트웨어 버전관리 도구 활용 능력 ○ 빌드 자동화 도구 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 제품소프트웨어에 변경의 필요성에 대해 이해하고자 하는 태도성 ○ 제품소프트웨어 버전관리의 중요성에 대한 이해도 ○ 제품의 품질 및 고객 우선주의 태도

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none">○ 제품소프트웨어의 특성에 대해 이해하고자 하는 태도○ 사용자 중심으로 제품소프트웨어 활용에 대해 이해하고자 하는 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위는 제품소프트웨어를 사용자에게 전달하기 위한 형태로 패키징하는 업무에 적용된다.
- 이 능력단위는 제품소프트웨어의 기획, 개선, 유지보수 목표에 따른 산출물들을 충실히 반영하고, 형상관리 지침과 매뉴얼 작성 지침을 기반으로 수행하여야 한다.
- 이 능력단위는 사용자가 제품소프트웨어에 대해 쉽게 이해하고 이를 활용할 수 있도록 사용자 편의성을 고려한 작업 수행이 이루어져야 한다.
- 이 능력단위의 적절한 수행을 위해서는 제품소프트웨어의 요구 환경과, 실행환경의 변화와 패키징용틀들의 변화에 대한 대응과 플랫폼별 패키징 표준에 대한 지속적인 훈련이 필요하다.

자료 및 관련 서류

- UML 작성표준
- 형상관리 지침
- 버전관리 도구 활용을 위한 매뉴얼과 버전, 리비전 부여 규칙
- 매뉴얼 작성 지침
- 표준 및 절차 매뉴얼
- 아키텍처 정의서
- 작업일지

장비 및 도구

- 서버, 데스크탑, 노트북, 프린터
- 네트워크, 인터넷
- 형상관리 프로그램
- 패키징 프로그램
- 문서 작성 프로그램
- 화면 녹화 프로그램
- 그래픽/동영상 편집 프로그램
- 복제방지, 저작권 표기 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 제품소프트웨어 패키지의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	
C. 서술형시험	√	√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	
G. 평가자 체크리스트		√
H. 피평가자 체크리스트		√
I. 일지/저널		
J. 역할연기	√	√
K. 구두발표		
L. 작업장평가	√	
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 제품소프트웨어 기획 의도에 대한 이해
 - 패키징 산출물 및 다이어그램 이해
 - 패키징시 필요한 패키징, 복제방지, 저작권 툴 등에 대한 이해
 - 형상관리 지침/도구에 대한 이해
 - 버전관리 규칙의 준수
 - 제품소프트웨어 UX/UI에 대한 이해
 - 제품소프트웨어 UX/UI와 일관된 매뉴얼 기술
 - 제품소프트웨어 실행 요구환경, 설치 환경에 대한 이해

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 의사표현능력, 기초외국어능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	정보능력	컴퓨터활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		수정
분류번호	기존	2001020209_15v3
	보완	2001020209_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v3
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		레벨 변경

분류번호 : 2001020215_15v3

능력단위 명칭 : 프로그래밍 언어 활용

능력단위 정의 : 프로그래밍 언어 활용이란 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020215_15v3.1 기본문법 활용하기	<p>1.1 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다.</p> <p>1.2 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.</p> <p>1.3 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 프로그래밍 언어 기본 문법○ 프로그래밍 언어 문법 오류○ 프로그래밍 언어 기능 구현을 위한 알고리즘○ 디버깅 기법○ 코드 검토 기법 <p>【기 술】</p> <ul style="list-style-type: none">○ 컴파일러 사용 능력○ 프로그램 언어 활용 능력○ IDE 도구 활용 능력○ 프로그램 디버깅 기술○ 프로그램 코드 검토 능력 <p>【태 도】</p> <ul style="list-style-type: none">○ 프로그래밍 언어의 규칙과 관습을 준수하려는 태도○ 문법 오류를 이해하고 수정하려는 적극적인 태도○ 기본 문법을 이해하고 활용하려는 적극적인 자세○ 기 작성된 코드를 분석하기 위한 치밀한 태도○ 코드 검토 결과를 수용하고 반영하려는 태도
2001020215_15v3.2 언어특성 활용하기	<p>2.1 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.</p> <p>2.2 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.</p> <p>2.3. 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용 할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ 절차적 프로그래밍 언어 ○ 객체지향 프로그래밍 언어 ○ 스크립트 언어 ○ 선언형 언어 ○ 디버깅 기법 ○ 코드 검토 기법 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 컴파일러 사용 능력 ○ 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 ○ 프로그램 디버깅 기술 ○ 프로그램 코드 검토 능력 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 프로그래밍 언어의 규칙과 관습을 준수하려는 태도 ○ 문법 오류를 이해하고 수정하려는 적극적인 태도 ○ 프로그래밍 언어의 특성을 적용하려는 태도 ○ 복잡한 구조 및 절차를 단순화하려는 의지 ○ 어플리케이션 구현의 방법과 절차를 문서화 하려는 의지
<p>2001020215_15v3.3</p> <p>라이브러리 활용하기</p>	<p>3.1 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.</p> <p>3.2 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다.</p> <p>3.3 선택 한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 라이브러리 기능에 대한 이해 ○ 애플리케이션 요구사항에 대한 이해 ○ 라이브러리 적용을 위한 기술(파일입출력, 데이터입출력, 예외 처리 등)에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 라이브러리 검색 능력 ○ 요구사항 구현에 필요한 라이브러리 선택 능력 ○ 라이브러리 설치 능력 ○ 라이브러리 적용 능력 ○ 코드 검토 능력 ○ 라이브러리를 활용한 프로토타입 작성 능력 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 프로그래밍 언어의 규칙과 관습을 준수하려는 태도 ○ 문법 오류를 이해하고 수정하려는 적극적인 태도 ○ 요구사항을 준수하려는 태도 ○ 라이브러리를 활용하는 적극적인 태도

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ 라이브러리를 모듈화하려는 의지 ○ 라이브러리의 영향범위를 최소화하려는 태도 ○ 라이브러리의 특성을 이해하려는 의지

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위에서는 프로그래밍 언어를 활용하는 부분으로 필요한 프로그래밍 언어를 선택하고 선택된 프로그래밍 언어 활용 지식과 기술을 학습한다.
- 프로그래밍 언어의 지식 및 기술과 연관된 도구들을 다룰 수 있는지에 대하여 고려되어야 한다.
- 이 능력단위에서는 선택된 프로그래밍 언어의 특성에 맞게 선택된 언어의 기본 문법, 언어의 특성, 언어의 라이브러리 활용 능력이 향상되도록 교육과 운용이 고려되어야 한다.

자료 및 관련 서류

- 프로그래밍 언어별 코딩 규칙(Rule)
- 프로그래밍 언어별 코딩 관습(Convention)
- 코딩 가이드(Coding Guide)
- 프로그래밍 언어별 사양서(Specification)

장비 및 도구

- 컴퓨터
- 빔프로젝터
- 화이트보드
- IDE 도구
- 프로그래밍 언어 개발환경 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 소프트웨어공학 활용의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험		√
D. 논술형시험		
E. 사례연구	√	√
F. 평가자 질문	√	
G. 평가자 체크리스트	√	√
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가시 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 프로그래밍 언어 활용에 대한 기본 소양 점검
 - 프로그래밍 언어 특성 활용 능력 점검
 - 프로그래밍 언어 라이브러리 적용을 위한 기본 능력 및 사고 점검
 - 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검

● 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

● 개발이력

구분		내용
직무명칭		응용SW엔지니어링
분류번호		2001020215_15v3
개발연도	현재	2015
	2차	정보통신산업진흥원
	최초(1차)	-
버전번호		v3(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2017~2019

분류번호 : 2001020216_15v3

능력단위 명칭 : 응용 SW 기초 기술 활용

능력단위 정의 : 응용 SW 기초 기술 활용이란 응용소프트웨어개발을 위하여 운영체제, 데이터베이스, 네트워크의 기초 기술을 적용하고 응용개발에 필요한 환경을 구축하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020216_15v3.1 운영체제 기초 활용하기	<p>1.1 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다.</p> <p>1.2 CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다.</p> <p>1.3 운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 각 운영체제 특징에 대한 이해○ Windows 운영체제 기본 명령어○ 리눅스 / 유닉스 계열 운영체제 기본 명령어○ 메모리 관리 기법의 이해○ 프로세스 스케줄링 기법의 이해○ 운영체제 환경변수 설정 방법 <p>【기 술】</p> <ul style="list-style-type: none">○ 운영체제 오류 해결방법 검색 기술○ 기본 명령어 활용 능력○ 각 운영체제에 맞는 응용프로그램 선택 능력○ 작업우선순위 변경 기술○ 운영체제 환경변수 설정 기술○ batch 파일, shell script 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none">○ 다양한 운영체제의 특징을 이해하려는 태도○ 운영체제 활용 중 발생한 문제를 해결하고자 하는 의지○ 운영체제 관련 기술 및 제품에 대한 정보수집에 대한 적극성○ 활용에 필요한 기술을 이해하려는 의지○ 콘솔/터미널을 활용하려는 태도
2001020216_15v3.2 데이터베이스 기초 활용하기	<p>2.1 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다.</p> <p>2.2 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.</p> <p>2.3 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ ER 다이어그램 작성 방법 ○ 무결성 제약조건의 이해 ○ 테이블 선언 및 조작용어 ○ 릴레이션의 특징 ○ DBMS 각 유형별 특징 ○ CRUD 연산에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 설계 모델링 기술 ○ 릴레이션 작성 기술 ○ E-R 다이어그램 작성 기술 ○ 테이블 제작 및 관리 언어 활용 능력 ○ 데이터베이스 개발 TOOL 사용 능력 ○ 오픈소스기반 DBMS 설치 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 정규화 과정을 지속적으로 관찰하는 태도 ○ 개발 수행에 요구되는 표준 절차를 준수하려는 태도 ○ 상황에 맞게 데이터베이스를 작성하고 변경하는 능동적인 태도 ○ 문제점을 해결하려는 의지 ○ 데이터베이스 관련 기술 및 제품에 대한 정보수집에 대한 적극성
2001020216_15v3.3 네트워크 기초 활용하기	<p>3.1 네트워크 계층구조에서 각 층의 역할을 설명할 수 있다.</p> <p>3.2 응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다.</p> <p>3.3 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none"> ○ 인터넷 구성의 개념 ○ 프로토콜 개념 ○ IP 주소 체계 ○ 트랜스포트 방식 개념 ○ TCP/UDP 방식 개념 ○ 네트워크 7 계층의 개념 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 필요 관련 정보 검색 기술 ○ 개발환경에 인터넷 구축 기술 ○ WINDOWS/UNIX/LINUX 인터넷 설정변경 기술 ○ 라우터 설정 기술 ○ 인터넷 환경변수 설정 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 적극적이고 세밀히 각 단계를 분석하는 자세 ○ 주어진 문제를 완수하는 책임감 ○ 네트워크 설정 관련 문제점을 해결하려는 의지 ○ 능동적 태도 ○ 네트워크 관련 기술 및 제품에 대한 정보수집에 대한 적극성
2001020216_15v3.4	<p>4.1 응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
기본 개발환경 구축하기	<p>4.2 응용개발에 필요한 개발도구를 설치하고 운용할 수 있다.</p> <p>4.3 웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용할 수 있다.</p>
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ 버전관리 시스템 구축 ○ 운영체제 설치 및 제거 방법 ○ 개발 프로그램 설치 방법 ○ 라이브러리 및 필요 패키지 설치 방법 ○ 필요 응용 소프트웨어에 대한 이해 ○ 사용자의 프로그램 실행 환경에 대한 이해 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 필요 프로그램 관련 정보 검색 기술 ○ 개발에 필요한 프로그래밍 언어 선택 능력 ○ 라이브러리 및 필요 패키지 선택 및 활용 능력 ○ 개발 TOOL에 맞는 운영체제 선택 능력 ○ 개발 TOOL 사용 능력 ○ 소프트웨어 버전관리 도구 활용 능력 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 사용할 프로그램의 특성을 이해하려는 태도 ○ 개발도구 설치시 발생한 문제점을 해결하려는 의지 ○ 자신이 수행한 작업에 대한 평가의 객관성 ○ 개발도구 관련 기술 및 제품에 대한 정보수집에 대한 적극성 ○ 오픈소스운동에 참여하려는 의지

◎ 적용범위 및 작업상황

고려사항

- 응용 SW 기초 기술 활용 능력을 습득하는데 있어 운영체제 기초 활용 능력은 우선적으로 고려되어야 할 능력이다.
- 기반서버라 함은 웹서버(아파치, nginx 등), 데이터베이스서버(MySQL, Postgresql 등), 프로그래밍 환경 (PHP 등) 등 응용 SW 개발에 활용이 되는 기본 소프트웨어를 의미한다.
- 운영체제 기초 활용하기와 기본 개발환경 구축하기는 실습 구성을 잘 고려하여야 한다.
- 데이터베이스기초활용 및 네트워크기초활용은 응용SW 개발에 필요한 최소한의 지식을 갖도록 구성된 능력단위요소임을 고려하여, 지식전달내용을 구성하여야 한다.

자료 및 관련 서류

- 오픈소스관련 사이트(opensource.org, www.gnu.org 등)
- Computer Science Curricular 2013 (<https://www.acm.org/education/CS2013-final-report.pdf>)

장비 및 도구

- 컴퓨터
- 빔프로젝터
- 프린터
- 화이트보드
- 오픈소스기반 가상머신 소프트웨어
- 오픈소스기반 기반서버(웹서버, DB서버, 프로그래밍환경)

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 소프트웨어공학 활용의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오		
C. 서술형시험		√
D. 논술형시험		
E. 사례연구		
F. 평가자 질문	√	
G. 평가자 체크리스트		√
H. 피평가자 체크리스트	√	
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 운영체제의 주요 기능, 구조, 특성에 대한 이해
 - 데이터베이스의 기본 개념 및 관계형데이터베이스 테이블 구성에 대한 이해
 - 네트워크의 기본 개념 및 OSI 7 Layer의 각 특성에 대한 이해
 - 기반서버의 주요 기능, 구조, 특성에 대한 이해
 - 오픈소스기반의 개발도구에 대한 이해

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 경청능력, 언어구사력, 기초 외국어 능력
2	문제해결능력	사고력, 문제처리능력
3	대인관계능력	팀웍능력, 리더십능력, 갈등관리능력, 협상능력, 고객서비스능력
4	정보능력	컴퓨터 활용능력, 정보처리능력
5	기술능력	기술이해능력, 기술선택능력, 기술적용능력

◎ 개발 이력

구 분		내 용
직무명칭		응용SW엔지니어링
분류번호		2001020216_15v3
개발연도	현재	2015
	2차	정보통신산업진흥원
	최초(1차)	-
버전번호		v3(보완)
개발자	현재	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2017~2019

분류번호 : 2001020228_16v4

능력단위 명칭 : 소프트웨어공학 활용

능력단위 정의 : 소프트웨어공학 활용이란 응용 소프트웨어 개발과 프로세스 적용활동의 관련된 지식을 소프트웨어의 완전성을 보장하고, 소프트웨어 품질을 평가하기 위해 CASE 도구와 형상관리를 통해 소프트웨어 공학 기술을 적용하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020228_16v4.1 CASE 도구 활용하기	<p>1.1 개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다.</p> <p>1.2 CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다.</p> <p>1.3 CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 소프트웨어 개발 방법론○ 프로젝트 관리 방법론○ 소프트웨어 개발 표준○ 요구공학 방법론○ 비용산정 모델○ 아키텍처 평가○ 소프트웨어 개발 프레임워크에 대한 지식 <p>【기 술】</p> <ul style="list-style-type: none">○ 협상 기술○ 의사결정 기술○ 프로세스 테일러링 기술○ UML 작성기술○ 모델링 기법○ 유즈케이스 모델 정의할 수 있는 기술○ 형상관리 기법 <p>【태 도】</p> <ul style="list-style-type: none">○ 새로운 공학도구를 사용하고 하는 도전의식○ 소프트웨어 개발방법론을 적용하기 위해 프로젝트를 분석하려는 자세○ 표준을 이해하고 준수하려는 의지○ 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도○ 소프트웨어의 품질에 대해 이해하고자 하는 태도
2001020228_16v4.2 품질요구사항 확인하기	<p>2.1 요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다.</p> <p>2.2 요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다.</p> <p>2.3 개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.</p>

능 력 단 위 요 소	수 행 준 거
	<p>【지 식】</p> <ul style="list-style-type: none"> ○ 소프트웨어 요구사항 명세 기준에 따른 품질 평가에 대한 지식 ○ 국제 표준 제품 품질특성에 대한 사항 ○ 품질 점검 기법(리뷰, 워크스루, 인스펙션 등) ○ 품질 관리 프로세스 ○ 소프트웨어 개발 표준 ○ 시정 조치 관리 프로세스 ○ 감사 프로세스 <p>【기 술】</p> <ul style="list-style-type: none"> ○ 리뷰 진행 기술 ○ 인스펙션 진행 기술 ○ 감사 기술 ○ 요구사항명세서 작성기술 ○ 소프트웨어 요구사항 품질 측정, 점검 기술 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 객관적인 평가방법을 작성하려는 객관성 ○ 품질요구사항에 대해 철저히 분석하려는 태도 ○ 품질요구사항에 대해 정확한 기준을 제시하려는 태도 ○ 품질 측정의 근거를 확보하기 위해 노력하는 태도 ○ 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위에서는 소프트웨어공학기술을 활용하여야 하는 부분으로 소프트웨어공학에 대한 기본지식을 갖추고 있어야 한다.
- 소프트웨어 개발방법론의 지식 및 기술과 연관된 도구들을 다룰 수 있는지에 대하여 고려되어야 한다.
- 품질평가를 하기 위하여 정확한 지표를 제시할 수 있는 객관적 측도를 선택하여 제시할 수 있는 교육이 요구된다.
- 이 능력단위에서는 소프트웨어의 특성을 분석하여 소프트웨어공학기술을 적용하여야 하기 때문에 소프트웨어의 다양한 특성을 학습할 수 있는 지속적인 교육과 운용이 고려되어야 한다.
- 이 능력단위는 능력단위의 직무에 따라 소프트웨어공학 활용과 SW개발방법론 활용으로 분할한다.

자료 및 관련 서류

- ISO/IEC 9126
- ISO/IEC 14598 - 소프트웨어 제품품질
- ISO 15504
- ISO/IEC 12207
- CMMI 표준

장비 및 도구

- 컴퓨터
- 빔프로젝터
- 프린터
- 화이트보드
- UML 도구
- CASE 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 소프트웨어공학 활용의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험		√
D. 논술형시험		√
E. 사례연구	√	√
F. 평가자 질문	√	
G. 평가자 체크리스트	√	
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 소프트웨어 공학기술에 대한 기본 소양 점검
 - 소프트웨어 개발방법론을 적용하기 위한 기본 능력 및 사고 점검
 - 관련 업무 영역의 개선 수행을 위한 소양 점검
 - 관련 업무 경험의 실효성/충분성 점검
 - 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 의사표현능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀웍능력, 협상능력
5	정보능력	컴퓨터활용능력, 정보처리능력

◎ 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020210_15v3
	보완	2001020228_16v4, 2001020229_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

분류번호 : 2001020229_16v4

능력단위 명칭 : 소프트웨어개발 방법론 활용

능력단위 정의 : 소프트웨어개발방법론 활용이란 응용소프트웨어 특성에 따라 정형화된 개발방법론을 선정하고, 프로젝트 특성에 맞도록 테일러링 하여 응용소프트웨어 개발에 활용하는 능력이다.

능 력 단 위 요 소	수 행 준 거
2001020229_16v4.1 SW 개발방법론 선정하기	<p>1.1 개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된 개발방법론 선정기준으로 활용할 수 있다.</p> <p>1.2 정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 소프트웨어 생명주기 모델○ 소프트웨어 개발 방법론○ 요구공학 방법론○ 비용산정 모델 <p>【기 술】</p> <ul style="list-style-type: none">○ 의사결정 기술○ 모델링 기법○ 모델링 도구 활용기술 <p>【태 도】</p> <ul style="list-style-type: none">○ 새로운 공학도구를 사용하고 하는 도전의식○ 소프트웨어 개발방법론을 적용하기 위해 프로젝트를 분석하려는 자세○ 소프트웨어의 품질에 대해 이해하고자 하는 태도
2001020229_16v4.2 SW개발방법론 테일러링하기	<p>2.1 프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다.</p> <p>2.2 확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다.</p> <p>2.3 응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다.</p> <p>2.4 테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다.</p> <p>【지 식】</p> <ul style="list-style-type: none">○ 반복, 점증 프로세스○ 프로젝트 관리 방법론○ 소프트웨어 개발 표준○ 비용산정 모델○ 소프트웨어 개발 프레임워크에 대한 지식 <p>【기 술】</p> <ul style="list-style-type: none">○ 협상 기술○ 의사결정 기술○ 프로세스 테일러링 기술

능 력 단 위 요 소	수 행 준 거
	<ul style="list-style-type: none"> ○ UML 작성기술 ○ 모델링 기법 ○ 유즈케이스 모델 정의할 수 있는 기술 ○ 형상관리 기법 <p>【태 도】</p> <ul style="list-style-type: none"> ○ 새로운 공학도구를 사용하고 하는 도전의식 ○ 표준을 이해하고 준수하려는 의지 ○ 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도 ○ 소프트웨어의 품질에 대해 이해하고자 하는 태도

◎ 적용범위 및 작업상황

고려사항

- 이 능력단위에서는 소프트웨어공학기술을 활용하여야 하는 부분으로 소프트웨어공학에 대한 기본지식을 갖추고 있어야 한다.
- 소프트웨어 개발방법론의 지식 및 기술과 연관된 도구들을 다룰 수 있는지에 대하여 고려되어야 한다.
- 이 능력단위에서는 소프트웨어의 특성을 분석하여 소프트웨어공학 기술을 적용하여야 하기 때문에 소프트웨어의 다양한 특성을 학습할 수 있는 지속적인 교육과 운용이 고려되어야 한다.
- 이 능력단위는 능력단위의 직무에 따라 소프트웨어공학 활용과 SW개발방법론 활용으로 분할한다.

자료 및 관련 서류

- ISO/IEC 14598
- ISO 15504
- ISO/IEC 12207
- CMMI 표준

장비 및 도구

- 컴퓨터
- 빔프로젝터
- 화이트보드
- UML 도구

재료

- 해당없음

◎ 평가지침

평가방법

- 평가자는 능력단위 소프트웨어공학 활용의 수행준거에 제시되어 있는 내용을 평가하기 위해 이론과 실기를 나누어 평가하거나 종합적인 결과물의 평가 등 다양한 평가 방법을 사용할 수 있다.
- 피 평가자의 과정평가 및 결과평가 방법

평 가 방 법	평 가 유 형	
	과 정 평 가	결 과 평 가
A. 포트폴리오		
B. 문제해결 시나리오	√	√
C. 서술형시험		√
D. 논술형시험		√
E. 사례연구	√	√
F. 평가자 질문	√	
G. 평가자 체크리스트	√	
H. 피평가자 체크리스트		
I. 일지/저널		
J. 역할연기		
K. 구두발표	√	√
L. 작업장평가		
M. 기타		

평가지 고려사항

- 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.
 - 소프트웨어 공학기술에 대한 기본 소양 점검
 - 소프트웨어 개발방법론을 적용하기 위한 기본 능력 및 사고 점검
 - 관련 업무 영역의 개선 수행을 위한 소양 점검
 - 관련 업무 경험의 실효성/충분성 점검
 - 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검

◎ 직업기초능력

순 번	직업기초능력	
	주요영역	하위영역
1	의사소통능력	문서이해능력, 문서작성능력, 의사표현능력
2	문제해결능력	사고력, 문제처리능력
3	자기개발능력	자아인식능력, 자기관리능력, 경력개발능력
4	대인관계능력	팀웍능력, 협상능력
5	정보능력	컴퓨터활용능력, 정보처리능력

◎ 개발이력

구분		내용
직무명칭		응용SW엔지니어링
능력단위 보완유형		분할
분류번호	기존	2001020210_15v3
	보완	2001020228_16v4, 2001020229_16v4
개발연도	현재	2016
	3차	2015
	2차	2014
	최초(1차)	-
버전번호		v4(보완)
개발자	현재	한국소프트웨어산업협회
	3차	한국소프트웨어산업협회
	2차	정보통신산업진흥원
	최초(1차)	-
향후 보완 연도(예정)		2019~2021
능력단위 보완사유		

3. 관련자격 개선의견

능 력 단 위	국 가 직 무 능 력 표 준 수 준	관 련 자 격	개 선 의 건
애플리케이션 요구사항 분석	7	정보관리기술사, 컴퓨터시스템응용기술사	없음
애플리케이션 리팩토링 인터페이스 설계 기능모델 설계 애플리케이션 설계 정적모델 설계 동적모델 설계 소프트웨어공학 활용 소프트웨어 개발방법론 활용	6	정보처리기사, 소프트웨어보안 진단원	없음
요구사항 확인 데이터 입출력 구현 통합 구현 정보시스템 이행 제품소프트웨어 패키징 서버프로그램 구현 인터페이스 구현 화면설계 애플리케이션 테스트관리	5	정보처리기사, 정보처리산업기사, 정보처리기능사	없음
애플리케이션 배포 프로그래밍 언어 활용 응용 SW 기초 기술 활용 화면 구현 애플리케이션 테스트 수행	3	정보처리산업기사, 정보처리기능사	없음

I. 정보기술개발 산업현장 검증

1 검증 사업체 현황(가나다순)

□ 응용SW엔지니어링

번 호	사 업 체 명	부 서	성 명
1	FavoriteMedium	엔지니어	-
2	KCC정보통신	선임	-
3	KG이니시스	대리	-
4	LGCNS	대리	-
5	LGCNS	과장	-
6	SK플래닛	매니저	-
7	공군군수전산소	주무관	-
8	김포대학교	교수	-
9	네오포인트	수석	-
10	대림아이앤에스	과장	-
11	두잇시스템	부장	-
12	디스플레이웍스	대표	-
13	디자인팩터	부장	-
14	디지털하우스	부장	-
15	레드비씨	차장	-
16	레드캡투어	부장	-
17	렉스켄	차장	-
18	리마코리아	대리	-
19	메가존	과장	-
20	미디어텍코리아	팀장	-
21	미소정보기술	책임	-
22	바를	대표	-
23	백석문화대학교	조교수	-
24	베네컴즈	팀장	-
25	삼성KPMG	부장	-

번 호	사 업 체 명	부 서	성 명
26	솔리데오시스템즈	차장	-
27	씨큐클라우드	대표	-
28	썬크포비엘	대리	-
29	아이온커뮤니케이션	팀장	-
30	아이투비컨설팅	대표	-
31	아프로시스	대표	-
32	앱센터	전문위원	-
33	에이더블유아이	수석	-
34	엔에이엔	이사	-
35	오메가정보시스템	대표	-
36	오보코리아	대표	-
37	옵니텔	본부장	-
38	웹투인	사원	-
39	위세아이텍	수석	-
40	인스피언컨설팅	차장	-
41	인터마블	대표	-
42	인피니트헬스케어	책임	-
43	지멘스	차장	-
44	지산웨어	부사장	-
45	케이씨에이	수석	-
46	케이에스인터랙티브	전무	-
47	케이엘넷	차장	-
48	토도스	대표	-
49	투에스아이티	부장	-
50	팬도메이트	대표	-
51	퓨전소프트	PMO	-
52	한국건물에너지기술원	본부장	-
53	한국보건복지정보개발원	차장	-
54	한국복지대학교	교수	-
55	한국인터넷진흥원	책임	-
56	한양	차장	-

※ 개인정보보호법에 근거하여 설문응답자의 성명은 기재하지 않음

2 검증 결과

□ 응용SW엔지니어링

구 분	세 부 내 용	업체수	평 균 점 수	평 가 결 과
1. 직무구조	1.1 능력단위 구성	56	4.1	적절
2. 직무 및 능력단위	2.1 직무정의	56	4.1	적절
	2.2 능력단위 정의	56	4.1	적절
	2.3 능력단위	56	4.1	적절
	2.4 능력단위 수준 평정	56	3.7	적절
3. 능력단위요소	3.1 능력단위요소	56	4.2	적절
	3.2 능력단위요소 수준 평정	56	3.9	적절
	3.3 수행준거	56	4.0	적절
	3.4 지식	56	4.0	적절
	3.5 기술	56	3.9	적절
	3.6 태도	56	3.9	적절
4. 직업기초능력	4.1 직업기초능력	56	4.1	적절
5. 적용범위 및 작업 상황	5.1 고려사항	56	4.2	적절
	5.2 자료·관련서류	56	3.9	적절
	5.3 장비·도구	56	3.7	적절
6. 평가지침	6.1 평가방법	56	4.0	적절
	6.2 평가시 고려사항	56	3.9	적절
7. 관련자격 개선 의견	7.1 자격과 표준 비교안	56	3.8	적절
	7.2 자격 개선의견	56	3.8	적절
8. 평생경력개발경로	8.1 산업통용 직책	56	3.8	적절
	8.2 수직/수평 경력이동	56	3.8	적절
	8.3 직무기술서	56	3.8	적절

Ⅱ. 정보기술개발 표준 개발 참여 전문가 명단

□ 응용SW엔지니어링

구 분		소 속	직 위	성 명
개발 전문가	산업현장	캡경영컨설팅	이사	강승훈
		유큐브	이사	김훈섭
		이씨오	팀장	박대우
		한국전자통신연구원	선임	이용희
		티에이코리아	이사	한기철
		엔가든	대표	홍성선
		한글과컴퓨터	책임	문성호
	교육 훈련	공주대학교	교수	안효범
		양영디지털고등학교	교사	이형우
		관동대학교	교수	김정아
	자격	밸류원컨설팅	책임	최현도
WG심의위원		다우기술	상무	어진선
		한국정보통신기술사협회	이사	김지은
		한국예탁결제원	차장	신우철
		송실대학교	교수	임용배
		네오포인트	이사	정동진
		알티베이스	부장	정현철
		한림성심대학교	교수	조규천
검토위원		포스코CT	차장	김문희
		한국외국어대학교	교수	김덕호
개발용역 수행기관		정보통신산업진흥원	팀장	이재덕
		정보통신산업진흥원	수석	조성현
		정보통신산업진흥원	책임	신민수
		CDC Group	상무	정준모
한국산업인력공단		표준개발실	실장	김록환
		표준개발3팀	팀장	양희경
		표준개발3팀	담당(차장)	안순건

※ **WG(Working Group)** : 협업형 개발체계를 구축하여 모든 직종을 대상으로 NCS를 개발하기 위하여 관련 전문기관, 부처별 추천 전문가, 교육훈련전문가, 자격전문가로 구성된 국가직무능력표준 분야별 개발 인력 풀

※ **WG(Working Group)심의위원회** : WG인력풀 중 산업현장, 교육·훈련, 자격전문가로 구성되며, NCS개발과 관련한 주요결정 심의, 표준안 사전심의 등 개발과정 전반에 걸친 질 관리를 담당하는 위원회

NCS 보완 참여 전문가 명단

□ 세분류명 : 응용SW엔지니어링

구 분		소 속	직 위	성 명
개발진	산업계	이비스툼	대표	강석진
		MDS테크놀로지	팀장	김중현
		와이즈스톤	책임	김영민
		레드비씨	부장	한성화
		KB데이터시스템	실장	이창근
		한국인터넷진흥원	책임	박양환
	교육훈련	(주)트리니티소프트	이사	이은진
	자격	베네컴스	대표	조준범
퍼실리테이터		한국소프트웨어산업협회	책임	변현욱
개발용역 수행기관	한국소프트웨어산업협회	부회장	서홍석	
	한국소프트웨어산업협회	책임	임하나	
	한국소프트웨어산업협회	책임	권태은	
	한국소프트웨어산업협회	선임	권오민	
한국산업인력공단	능력평가이사		이 사	김 시 태
	NCS및신직업자격안정화추진단	단 장	양 기 훈	
		부 단 장	장 덕 호	
		팀 장	강 석 주	
		차 장	권 동 호	
		차 장	유 지 용	
		선임연구원	김 승 수	
		전문위 원	조 우 형	
		전문위 원	우 미 혜	
		대 리	김 은 지	
		대 리	박 준 범	
		전문위 원	황 수 영	
		직 원	이 현 욱	
		직 원	김 재 은	
		직 원	김 준 현	

1 평생경력개발 경로

① 개발목적

- 산업현장의 근로자를 경력개발, 채용·승진 등 인사관리를 위하여 국가직무능력표준에 따라 경력개발경로 콘텐츠* 개발

* 국가직무능력표준 개발시 평생경력개발경로 모형, 직무기술서, 채용·배치·승진 체크리스트, 자가진단도구 개발

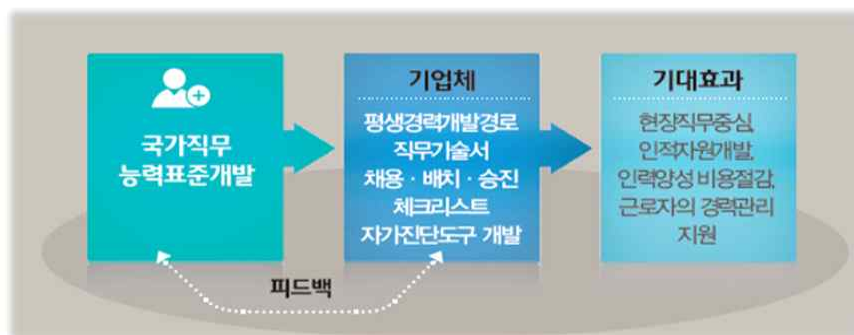
② 활용대상

활용콘텐츠 개발	활용대상
평생경력개발경로 모형	사업체, 근로자
직무기술서	사업체
채용·배치·승진 체크리스트	사업체
자가진단도구	근로자

③ 활용방법

- 평생경력개발 콘텐츠의 내용과 사업체의 경력개발경로, 직무기술서 등을 비교·분석
- 평생경력개발 콘텐츠를 그대로 활용하거나 변형하여 활용
 - 콘텐츠의 내용이 사업체의 경력개발경로 등이 유사한 경우에는 그대로 개발된 콘텐츠를 그대로 활용
 - 콘텐츠의 내용이 사업체와 일부 상이한 경우에는 사업체의 특성에 맞게 콘텐츠의 내용을 변경하여 활용

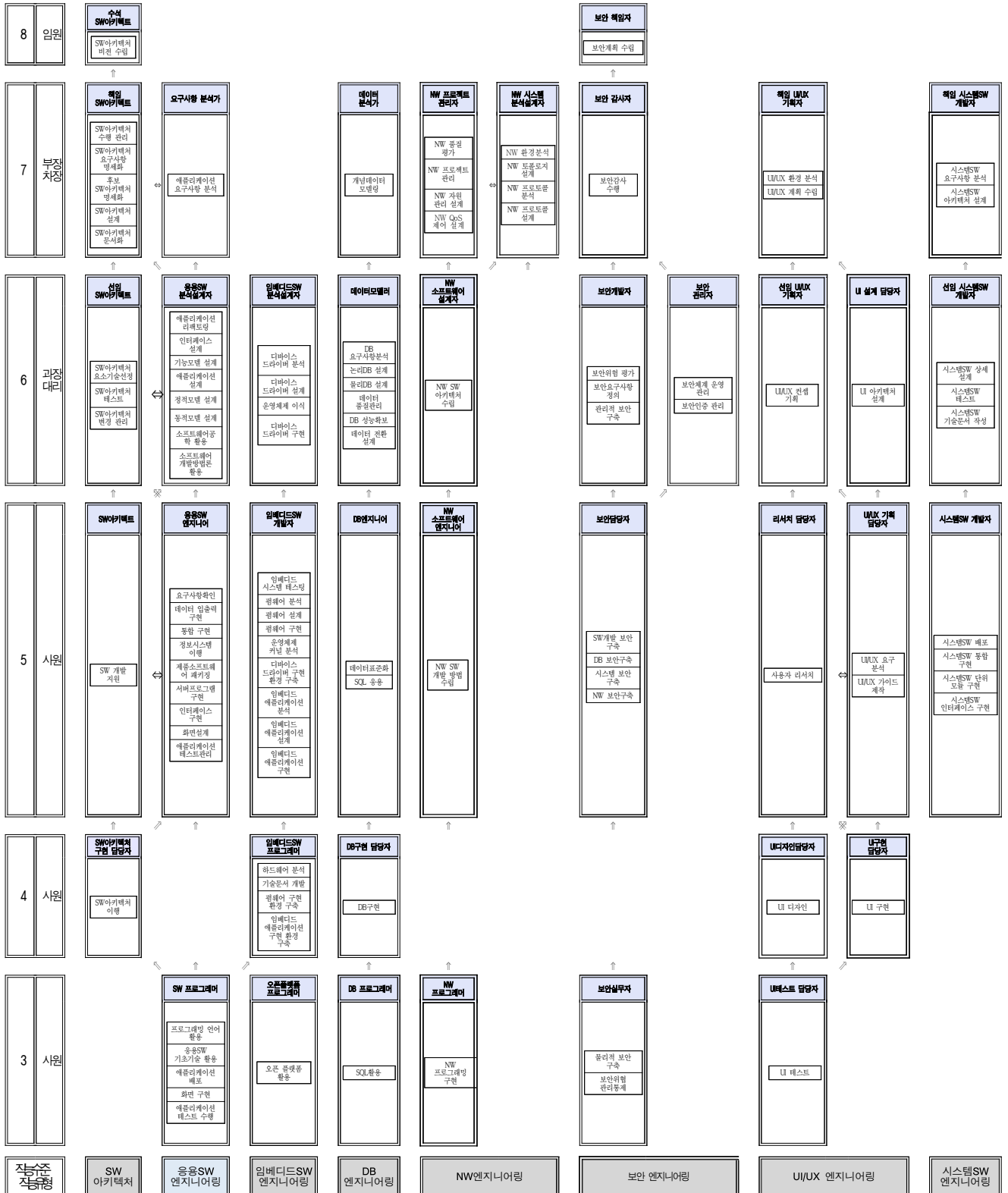
④ 기대효과



1-1. 능력단위 구조도

8	SW아키텍처 비전 수립	보안계획 수립						
7	SW아키텍처 수행 관리 SW아키텍처 요구사항 명세화 후보 SW아키텍처 평선회 SW아키텍처 설계 SW아키텍처 문서화	애플리케이션 요구사항 분석		개념데이터 모델링	NW 환경분석 NW 토폴로지 설계 NW 프로토콜 분석 NW 프로토콜 설계 NW 품질 평가 NW 프로젝트 관리 NW 자원 관리 설계 NW QoS 제어 설계	보안감사 수행	UI/UX 계획 수립 UI/UX 환경 분석	시스템SW 요구사항 분석 시스템SW 아키텍처 설계
6	SW아키텍처 요소기술선정 SW아키텍처 테스트 SW아키텍처 변경 관리	애플리케이션 리팩토링 인터페이스 설계 기능모델 설계 애플리케이션 설계 정적모델 설계 동적모델 설계 소프트웨어공학 활용 소프트웨어 개발방법론 활용	디바이스 드라이버 분석 디바이스 드라이버 설계 운영체제 이식 디바이스 드라이버 구현	DB 요구사항분석 논리DB 설계 물리DB 설계 데이터품질관리 DB 성능확보 DB 전환설계	NW SW 아키텍처 수립	보안체계 운영 관리 보안인증 관리 보안위협 평가 보안요구사항 정의 관리적 보안 구축	UI/UX 컨셉 기획 UI 아키텍처 설계	시스템SW 상세 설계 시스템SW 테스트 시스템SW 기술문서 작성
5	SW 개발 지원	요구사항 확인 데이터 입출력 구현 통합 구현 정보시스템 이행 제품소프트웨어 패키징 서버프로그램 구현 인터페이스 구현 화면설계 애플리케이션 테스트관리	임베디드 시스템 테스팅 펌웨어 분석 펌웨어 설계 펌웨어 구현 운영체제 커널 분석 디바이스 드라이버 구현 환경 구축 임베디드 애플리케이션 분석 임베디드 애플리케이션 설계 임베디드 애플리케이션 구현	데이터표준화 데이터전환 SQL응용	NW SW 개발 방법 수립	SW개발 보안 구축 데이터베이스 보안 구축 시스템 보안 구축 네트워크 보안 구축	사용자 리서치 UI/UX 요구 분석 UI/UX 가이드 제작	시스템SW 배포 시스템SW 통합 구현 시스템SW 단위 모듈 구현 시스템SW 인터페이스 구현
4	SW아키텍처 이행		하드웨어 분석 기술문서 개발 펌웨어 구현 환경 구축 임베디드 애플리케이션 구현 환경 구축	DB구현			UI 디자인 UI 구현	
3		애플리케이션 배포 프로그래밍 언어 활용 응용 SW 기초 기술 활용 화면 구현 애플리케이션 테스트 수행	오픈 플랫폼 활용	SQL활용	NW 프로그래밍 구현	물리적 보안 구축 보안위협 관리통제	UI 테스트	
직 능 수 준 / 직 능 유 형	SW아키텍처	응용SW 엔지니어링	임베디드SW 엔지니어링	DB엔지니어링	NW엔지니어링	보안엔지니어링	UI/UX 엔지니어링	시스템SW 엔지니어링

1-2. 평생경력개발 체계도



1-3. 평생경력개발경로



2.1. 직무기술서 개요

○ 개념 : 직무기술서는 해당 직무의 목적과 업무의 범위, 주요 책임, 요구받는 역할, 직무 수행 요건 등 직위에 관한 정보를 제시한 문서를 의미

○ 구성요소

- 직무, 능력단위분류번호, 능력단위, 직무목적, 직무 책임 및 역할, 직무수행요건으로 구성
- 추가 정보 제공을 위해 개발 날짜, 개발 기관을 추가 제시

구 성 요 소	세 부 내 용
능력단위분류번호	<ul style="list-style-type: none"> 전체 직무 구조 관리를 위한 직무 고유의 코드번호
능력단위	<ul style="list-style-type: none"> 수행하고자 하는 능력단위의 명칭
직무목적	<ul style="list-style-type: none"> 직무를 수행함으로써 이루고자 하는 직무의 목적
개발날짜	<ul style="list-style-type: none"> 개발된 년, 월, 일
개발기관	<ul style="list-style-type: none"> 직무기술서를 개발한 기관
직무 책임 및 역할	<ul style="list-style-type: none"> 직무에 대한 책임 및 역할 영역 분류 및 상세 내용
직무수행요건	<ul style="list-style-type: none"> 직무를 수행하기 위하여 개인이 일반적으로 갖추어야 할 사항 <ul style="list-style-type: none"> 학력, 자격증, 지식 및 스킬, 사전 직무경험, 직무숙련기간 등

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020201_16v3
		능 력 단 위	요구사항 확인
직무 목적	업무 분석가가 수집·분석·정의한 요구사항과 이에 따른 분석모델에 대해서 확인과 현행 시스템에 대한 분석할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	정보통신산업진흥원

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
현행 시스템 분석하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다. ○ 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.
요구사항 확인하기	<ul style="list-style-type: none"> ○ 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다. ○ 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다. ○ 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.
분석모델 확인하기	<ul style="list-style-type: none"> ○ 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다. ○ 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다. ○ 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 4년제 대학 졸업	(전공: 전산 관련학, 경영정보 관련학)
	· SW기획 과정	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보기술프로젝트관리전문가 	

지식·기술	<ul style="list-style-type: none"> · 가상화 관련 지식 · 가상화 관련 기술 · 내부 및 외부 환경 분석 기술 · 네트워크 구성 및 관리 능력 · 리뷰 진행 기술 · 분석 CASE 도구 사용 기술 · 소프트웨어 개발 방법론 · 저장장치 구성 및 관리 능력 · 업무 특성에 대한 이해 · 요구공학 방법론 · 요구분석기법 · 요구사항 관리 도구 사용 기술 · 운영체제 구성 및 관리 기술 · 유즈케이스 작성 능력 · 클라우드 컴퓨팅 관련 지식 · 타당성 분석기법 · 통계학 · 프로젝트 환경 및 특수성 · 프로젝트 환경 및 특수성 · 플랫폼에 따른 기능 및 성능 특성 · 해당 산업 분야에 대한 지식 · 해당 플랫폼에 대한 지식 · DBMS 구성 및 관리 기술 · UML 작성 기술
사전직무경험	<ul style="list-style-type: none"> · 개발자 테스트 · 정보시스템 이행
직무숙련기간	<ul style="list-style-type: none"> · 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020205_16v4
		능 력 단 위	데이터 입출력 구현
직무 목적	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 구현할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	정보통신산업진흥원

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
논리 데이터저장소 확인하기	<ul style="list-style-type: none"> ○ 업무 분석가, 데이터베이스 엔지니어가 작성한 논리 데이터저장소 설계 내역에서 정의된 데이터의 유형을 확인하고 식별할 수 있다. ○ 논리 데이터저장소 설계 내역에서 데이터의 논리적 단위와 데이터 간의 관계를 확인할 수 있다. ○ 논리 데이터저장소 설계 내역에서 데이터 또는 데이터간의 제약조건과 이들 간의 관계를 식별할 수 있다.
물리 데이터저장소 설계하기	<ul style="list-style-type: none"> ○ 논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을 반영한 물리 데이터저장소 설계를 수행할 수 있다. ○ 논리 데이터저장소 설계를 바탕으로 목표 시스템의 데이터 특성을 반영하여 최적화된 물리 데이터저장소를 설계할 수 있다. ○ 물리 데이터저장소 설계에 따라 데이터저장소에 실제 데이터가 저장될 물리적 공간을 구성할 수 있다.
데이터 조작 프로시저 작성하기	<ul style="list-style-type: none"> ○ 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소에 연결을 수행하는 프로시저를 작성할 수 있다. ○ 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소로부터 데이터를 읽어 오는 프로시저를 작성할 수 있다. ○ 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 변경 내용 또는 신규 입력된 데이터를 데이터 저장소에 저장하는 프로시저를 작성할 수 있다. ○ 구현된 데이터 조작 프로시저를 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.
데이터 조작 프로시저 최적화하기	<ul style="list-style-type: none"> ○ 프로그래밍 언어와 도구에 대한 이해를 바탕으로 응용소프트웨어 설계, 물리 데이터저장소 설계와 운영 환경을 고려하여 데이터 조작 프로시저의 성능을 예측할 수 있다. ○ 업무 분석가에 의해 정의된 요구사항을 기준으로, 성능측정 도구를 활용하여 데이터 조작 프로시저의 성능을 측정할 수 있다. ○ 실 데이터를 기반으로 테스트를 수행하여 데이터 조작 프로시저의 성능에 영향을 주는 병목을 파악할 수 있다. ○ 테스트 결과와 정의된 요구사항을 기준으로 데이터조작 프로시저의 성능에 따른 이슈

주요업무	책임 및 역할
	발생 시 이에 대해 해결할 수 있다.

□ 직무수행 요건

구분	상세내용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업 (전공: 전산 관련학)
	<ul style="list-style-type: none"> • SW개발 과정, DB구현 과정 (분야: IT서비스, IT기술)
자격증	<ul style="list-style-type: none"> • 정보처리기사 • 정보처리산업기사 • 정보처리기능사
지식·기술	<ul style="list-style-type: none"> • 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 • 구현에 요구되는 아키텍처의 이해 • 단위테스트 도구 활용기법 • 데이터베이스에 대한 이해 • 모델링 검증 도구 활용 능력 • 설계 산출물의 이해 • 성능 측정 도구 활용 능력 • 성능 평가 기준에 대한 이해 • 소스코드 인스펙션 능력 • 소스코드 표준 문서화 능력 • 소프트웨어 테스트에 대한 이해 • 수학, 통계 지식 • 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등) • 자료구조 • 컴파일러 사용법 • 쿼리(Query) 성능 측정 도구 활용 능력 • 트랜잭션 인터페이스 설계 능력 • 표준 단어, 용어, 도메인, 코드에 대한 정의 능력 • 프로그래밍 표준에 대한 이해 • 프로그램 디버깅 기법 • 프로그래밍언어(C, C++, Java-JSP/Servlet 등)와 도구(IDE 등)의 활용방법에 대한 이해 • DBMS 관리 도구 활용 능력 • DBMS 사용 기술 • E-R Modeling • E-R 모델링 도구 활용능력 • IDE 도구 활용 능력 • ORM 프레임워크 활용 능력 • RDBMS를 통한 구현 능력 • SQL(Structure Query Language) • UML 모델링 도구 활용 능력
사전직무경험	<ul style="list-style-type: none"> • 개발자 테스트 • 정보시스템 이행

직무숙련기간	· 약 2년
--------	--------

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020206_16v4
		능 력 단 위	통합 구현
직무 목적	모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하여 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	정보통신산업진흥원

□ 직무 책임 및 역할

주 요 업 무	책 임 및 역 할
연계 데이터 구성하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다. ○ 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다. ○ 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.
연계 메커니즘 구성하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다. ○ 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다. ○ 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다. ○ 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.
내외부 연계 모듈 구현하기	<ul style="list-style-type: none"> ○ 구성된 연계 메커니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다. ○ 구성된 연계 메커니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다. ○ 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다. ○ 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 4년제 대학 졸업	(전공: 전산 관련학)

	<ul style="list-style-type: none"> · SW설계 과정, SW개발 과정 	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보관리기술사 · 컴퓨터시스템응용기술사 · 데이터아키텍처전문가 · 정보기술프로젝트관리전문가 	
지식·기술	<ul style="list-style-type: none"> · 개발 방법론 이해 · 구현에 요구되는 아키텍처의 이해 · 기술영역별 미들웨어/솔루션 활용 · 네트워크 이해 · 단위테스트 도구 활용기법 · 데이터 보안 기술 · 데이터 연계/이관 도구 활용 능력 · 데이터베이스 이해 · 분산 프로그래밍에 대한 이해 · 설계 모델링 기법 · 설계 산출물의 이해 · 설계 모델링 기술 · 소프트웨어 아키텍처에 대한 이해 · 소프트웨어 테스트에 대한 이해 · 업무 특성에 대한 이해 · 프로그래밍어 활용방법의 이해 · 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 · 프로그램 디버깅 기법 · 형상관리 도구 활용 기법 · 해당 산업 분야에 대한 지식 · IDE 도구 활용 능력 · IDE 및 개발환경 도구 활용 · UML 작성 기술 	
사전직무경험	<ul style="list-style-type: none"> · 요구사항 확인 · 데이터입출력 구현 	
직무숙련기간	<ul style="list-style-type: none"> · 약 2년 	

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020208_16v3
		능 력 단 위	정보시스템 이행
직무 목적	개발자 환경에서 개발한 결과물을 운영 환경에 설치하고, 사용자 요구사항과 최종적으로 일치하는지에 대해 승인을 얻어 응용소프트웨어 결과물을 사용자에게 전달하여 인계하고 시스템을 운영할 수 있도록 교육하고 지원할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	정보통신산업진흥원

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
정보시스템 사용자 교육하기	<ul style="list-style-type: none"> ○ 개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다. ○ 개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다. ○ 효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검 할 수 있다. ○ 교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육결과 및 성과를 평가 할 수 있다.
정보시스템 이행하기	<ul style="list-style-type: none"> ○ 하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다. ○ 시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발 할 수 있다. ○ 정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증 할 수 있다. ○ 정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.
정보시스템 안정화하기	<ul style="list-style-type: none"> ○ 정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다. ○ 정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정보시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다. ○ 개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 2년제 대학 졸업	(전공: 전산 관련학)

	<ul style="list-style-type: none"> · SW개발 과정 	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보처리기능사 	
지식·기술	<ul style="list-style-type: none"> · 국제 표준 제품 품질특성에 대한 사항 · 결함관리 도구 활용 능력 · 교육성과 분석 기법 · 교육성과 측정 기법 · 멀티미디어 작성 도구 활용 능력 · 문서작성 도구 활용 방법 · 문서작성 도구 활용 능력 · 성능 측정 도구 활용 능력 · 소프트웨어 아키텍처에 대한 이해 · 시스템 성능 분석 및 진단에 대한 지식 · 시스템 운영 환경에 대한 이해 · 시스템 운영에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 · 애플리케이션 모니터링 도구 활용 능력 · 애플리케이션 배포 도구 활용 능력 · 애플리케이션 패키징 기술 · 업무 특성에 대한 이해 · 표준 양식(Template) 작성 및 활용법 · 표준 용어집 · 피교육 집단 특성에 대한 이해 · 프리젠테이션 기법 · 해당 산업 분야에 대한 지식 · 형상관리 도구 활용 기법 	
사전직무경험	<ul style="list-style-type: none"> · 제품소프트웨어 패키징 	
직무숙련기간	<ul style="list-style-type: none"> · 약 1년 	

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020209_16v4
		능 력 단 위	제품소프트웨어 패키징
직무 목적	개발이 완료된 제품소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전관리를 수행할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	정보통신산업진흥원

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
제품소프트웨어 패키징하기	<ul style="list-style-type: none"> ○ 신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다. ○ 이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다. ○ 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.
제품소프트웨어 매뉴얼 작성하기	<ul style="list-style-type: none"> ○ 사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. ○ 사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. ○ 사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.
제품소프트웨어 버전관리하기	<ul style="list-style-type: none"> ○ 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다. ○ 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다. ○ 버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 2년제 대학 졸업 (전공: 전산 관련학)
	· SW개발 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보처리기능사

지식·기술	<ul style="list-style-type: none"> · 각종 매뉴얼 작성법 · 결함관리 도구 활용 능력 · 국제 표준 제품 품질특성에 대한 사항 · 릴리즈 노트 작성법 · 멀티미디어 작성 도구 활용 능력 · 문서작성 도구 활용 능력 · 빌드 자동화 도구 활용 능력 · 사용자의 제품 실행 환경에 대한 이해 · 성능 측정 도구 활용 능력 · 소프트웨어 버전관리 도구 활용 능력 · 소프트웨어 버전관리 및 도구 사용에 대한 이해 · 암호화/보안 모듈 (DRM) 탑재 기술 · 애플리케이션 모니터링 도구 활용 능력 · 애플리케이션 배포 도구 활용 능력 · 애플리케이션 패키징 기술 · 장애 복구 처리 절차 · 제품 패키징, 배포 관련 표준에 대한 지식 · 제품소프트웨어 관련 시장에 대한 지식 · 제품소프트웨어 사용에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 · 제품소프트웨어 아키텍처에 대한 이해 · 제품소프트웨어가 적용되는 업무 특성에 대한 이해 · 표준 용어집 · 프리젠테이션 기법 · 협업도구 활용 능력 · 형상관리 도구 활용 기법 · API 문서 생성 기술(JavaDoc, doxygen) · Help File 생성 기술
사전직무경험	<ul style="list-style-type: none"> · 화면 구현
직무숙련기간	<ul style="list-style-type: none"> · 약 1년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020211_16v4
		능 력 단 위	서버프로그램 구현
직무 목적	서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책 임 및 역 할
개발환경 구축하기	<ul style="list-style-type: none"> ○ 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행한다. ○ 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축한다. ○ 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축한다.
공통 모듈 구현하기	<ul style="list-style-type: none"> ○ 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성한다. ○ 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현한다. ○ 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 한다.
서버 프로그램 구현하기	<ul style="list-style-type: none"> ○ 업무 프로세스 맵과 세부 업무 프로세스를 확인한다. ○ 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현한다. ○ 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현한다. ○ 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행한다.
배치 프로그램 구현하기	<ul style="list-style-type: none"> ○ 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현한다. ○ 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현한다. ○ 개발된 배치 프로그램을 테스트를 수행한다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<div> <div>· 4년제 대학 졸업</div> <div>(전공: 전산 관련학)</div> </div>

	· SW개발 과정, DB구현 과정 (분야: IT서비스, IT기술)
자 격 증	· 정보처리기사 · 정보보안기사 · 소프트웨어 보안약점 진단원 · 정보처리산업기사 · 정보보안산업기사
지식·기술	· IDE 도구 활용 능력 · 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 · 개발에 필요한 하드웨어 및 소프트웨어 사양 · 개발환경에 요구되는 아키텍처 · 결함 관리 · 구현에 요구되는 아키텍처 · 네트워크 개념 · 네트워크 활용 기술 · 단위 업무 · 단위테스트 기술동향 및 도구 · 단위테스트 도구 활용기법 · 데이터베이스 개념 · 모듈화 기법 · 보안 관련 개념 · 보안도구 활용 능력 · 분산 처리 개념 · 분산 프로그래밍 개념 · 서버 개발 프레임워크 개념 · 설계 산출물 해석방법 · 소프트웨어 테스트 · 인터페이스 도구/기법 (API, EAI, WebService, 등) · 재사용 기법 · 컴파일러 사용 능력 · 컴파일러 사용법 · 프레임워크 활용 기술 · 프로그래밍언어 활용 기술 · 프로그램 디버깅 기술 · 프로그램 언어 활용방법 · 프로그램 코드 검토 기법 · 프로그램 코드 검토 기술 · 협업도구 활용 능력 · 형상관리 도구 활용 기법 · 형상관리 도구 활용 능력
사전직무경험	· 프로그래밍 언어 활용 · 응용 SW기초 기술 활용
직무숙련기간	· 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020212_16v4
		능 력 단 위	인터페이스 구현
직무 목적	인터페이스 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책 임 및 역 할
인터페이스 설계서 확인하기	<ul style="list-style-type: none"> ○ 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인한다. ○ 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별한다. ○ 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인한다.
인터페이스 기능 구현하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의한다. ○ 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현한다. ○ 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의한다. ○ 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용한다.
인터페이스 구현 검증하기	<ul style="list-style-type: none"> ○ 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비한다. ○ 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인한다. ○ 인터페이스 오류처리 사항을 확인하고 보고서를 작성한다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업 (전공: 전산 관련학)
	<ul style="list-style-type: none"> • SW개발 과정, 네트워크 프로그래밍 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> • 정보처리기사 • 정보보안기사 • 정보통신기사 • 정보처리산업기사 • 정보보안산업기사

	<ul style="list-style-type: none"> · 정보통신산업기사
지식·기술	<ul style="list-style-type: none"> · IDE 도구 활용 능력 · IDE 및 개발환경 도구 활용 · UML 작성 기술 · 개발 방법론 · 구현에 요구되는 아키텍처 · 기술영역별 미들웨어/솔루션 활용 · 기술영역별 미들웨어/솔루션 활용 기술 · 네트워크 · 단위테스트 도구 활용기법 · 데이터 보안 기술 · 데이터 연계/이관 도구 활용 능력 · 데이터베이스 · 디자인 패턴 · 분산 프로그래밍 · 산업 분야별 업무특성 · 설계 모델링 기법 · 설계 모델링 기술 · 설계 산출물 · 소프트웨어 아키텍처 · 소프트웨어 테스트 · 프레임워크 활용 능력 · 프로그램 디버깅 기법 · 프로그램 언어 활용 능력 · 프로그램 코드검토 기법 · 형상관리 도구 활용 기법
사전직무경험	<ul style="list-style-type: none"> · 프로그래밍 언어 활용 · 응용 SW기초 기술 활용
직무숙련기간	<ul style="list-style-type: none"> · 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020214_16v4
		능 력 단 위	애플리케이션 배포
직무 목적	애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포 할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
애플리케이션 배포 환경 구성하기	<ul style="list-style-type: none"> ○ 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다. ○ 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다. ○ 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다. ○ 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.
플리케이션 소스 검증하기	<ul style="list-style-type: none"> ○ 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다. ○ 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다. ○ 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.
애플리케이션 빌드하기	<ul style="list-style-type: none"> ○ 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다. ○ 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다. ○ 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다. ○ 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지 여부를 확인할 수 있다. ○ 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명할 수 있다.
애플리케이션 배포하기	<ul style="list-style-type: none"> ○ 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다. ○ 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다. ○ 애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다. ○ 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 2년제 대학 졸업	(전공: 전산 관련학)

	<ul style="list-style-type: none"> · 소프트웨어 배포 자동화과정 	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 	
지식·기술	<ul style="list-style-type: none"> · 배포 도구 명령어 · 배포 스크립트 작성 기술 · 빌드 검증 기술 · 빌드 결과 분석 지식 · 빌드 도구 명령어 · 빌드 스크립트 작성 기술 · 빌드 실행 기술 · 소스코드 검증 기법 · 소스코드 검증 도구 결과 분석 기술 · 소스코드 검증 도구 사용 기술 · 소스코드 동적 검증 분석 기법 · 소스코드 정적 검증 분석 기법 · 애플리케이션 빌드 도구 적용 기술 · 애플리케이션 빌드 스크립트 문법 · 애플리케이션 빌드 절차 · 애플리케이션 운영 플랫폼 정보 · 애플리케이션 원복 실행 도구 사용법 · 애플리케이션 원복 절차 · 운영 배포 결과 확인 기법 · 형상관리 지침 · 환경 설정 기법 · 배포 도구 설치 및 사용법 · 빌드 도구 설치 및 사용법 · 애플리케이션 배포 환경 구성 지식 · 형상관리 도구 설치 및 사용법 · 소스코드 확보 기술 · 애플리케이션 빌드 스크립트 작성 기법 · 애플리케이션 빌드 시스템 가이드 · 애플리케이션 운영 환경 배포 기술 · 애플리케이션 운영 환경 정보 · 애플리케이션 빌드(Build) 절차 · 형상관리 도구 명령어 	
사전직무경험	<ul style="list-style-type: none"> · 프로그래밍 언어활용 · 응용SW기초기술활용 	
직무숙련기간	<ul style="list-style-type: none"> · 약 1년 	

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020215_15v3
		능 력 단 위	프로그래밍 언어 활용
직무 목적	프로그래밍 언어 활용이란 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현할 수 있다.		
개발 날짜	2015. 11. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
기본문법 활용하기	<ul style="list-style-type: none"> ○ 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용한다. ○ 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용한다. ○ 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용한다.
언어특성 활용하기	<ul style="list-style-type: none"> ○ 프로그래밍 언어별 특성을 파악하고 설명한다. ○ 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현한다. ○ 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용한다.
라이브러리 활용하기	<ul style="list-style-type: none"> ○ 애플리케이션에 필요한 라이브러리를 검색하고 선택한다. ○ 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성한다. ○ 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용한다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 고등학교 졸업 (전공: 소프트웨어학)
	<ul style="list-style-type: none"> • 프로그래밍 언어 활용과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> • 정보처리기사 • 정보처리산업기사 • 정보처리기능사
지식·기술	<ul style="list-style-type: none"> • 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 • 객체지향 프로그래밍 언어

	<ul style="list-style-type: none"> • 디버깅 기법 • 라이브러리 검색 능력 • 라이브러리 기능에 대한 이해 • 라이브러리 설치 능력 • 라이브러리 적용 능력 • 라이브러리 적용을 위한 기술(파일입출력, 데이터입출력, 예외 처리 등)에 대한 이해 • 라이브러리를 활용한 프로토타입 작성 능력 • 선언형 언어 • 스크립트 언어 • 애플리케이션 요구사항에 대한 이해 • 요구사항 구현에 필요한 라이브러리 선택 능력 • 절차적 프로그래밍 언어 • 컴파일러 사용 능력 • 코드 검토 기법 • 코드 검토 능력 • 프로그래밍 언어 기능 구현을 위한 알고리즘 • 프로그래밍 언어 기본 문법 • 프로그래밍 언어 문법 오류 • 프로그램 디버깅 기술 • 프로그램 언어 활용 능력 • 프로그램 코드 검토 능력 • IDE 도구 활용 능력
사전직무경험	<ul style="list-style-type: none"> • 해당사항없음
직무숙련기간	<ul style="list-style-type: none"> • 약 1년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020216_15v3
		능 력 단 위	응용 SW 기초 기술 활용
직무 목적	응용소프트웨어개발을 위하여 운영체제, 데이터베이스, 네트워크의 기초 기술을 적용하고 응용개발에 필요한 환경을 구축할 수 있다.		
개발 날짜	2015. 11. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
운영체제 기초 활용하기	<ul style="list-style-type: none"> ○ 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명한다. ○ CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용한다. ○ 운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정한다.
데이터베이스 기초 활용하기	<ul style="list-style-type: none"> ○ 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정한다. ○ 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의한다. ○ 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명한다.
네트워크 기초 활용하기	<ul style="list-style-type: none"> ○ 네트워크 계층구조에서 각 층의 역할을 설명한다. ○ 응용의 특성에 따라 TCP와 UDP를 구별하여 적용한다. ○ 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명한다.
기본 개발환경 구축하기	<ul style="list-style-type: none"> ○ 응용개발을 위하여 선정된 운영체제를 설치하고 운용한다. ○ 응용개발에 필요한 개발도구를 설치하고 운용한다. ○ 웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용한다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 고등학교 졸업 (전공: 소프트웨어학)
	· 정보처리 기초 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보처리기능사

지식·기술	<ul style="list-style-type: none"> · batch 파일, shell script 활용 능력 · CRUD 연산에 대한 이해 · DBMS 각 유형별 특징 · E-R 다이어그램 작성 기술 · ER 다이어그램 작성 방법 · IP 주소 체계 · TCP/UDP 방식 개념 · Windows 운영체제 기본 명령어 · WINDOWS/UNIX/LINUX 인터넷 설정변경 기술 · 각 운영체제 특징에 대한 이해 · 각 운영체제에 맞는 응용프로그램 선택 능력 · 개발 TOOL 사용 능력 · 개발 TOOL에 맞는 운영체제 선택 능력 · 개발 프로그램 설치 방법 · 개발에 필요한 프로그래밍 언어 선택 능력 · 개발환경에 인터넷 구축 기술 · 기본 명령어 활용 능력 · 네트워크 7 계층의 개념 · 데이터베이스 개발 TOOL 사용 능력 · 라우터 설정 기술 · 라이브러리 및 필요 패키지 선택 및 활용 능력 · 라이브러리 및 필요 패키지 설치 방법 · 리눅스 / 유닉스 계열 운영체제 기본 명령어 · 릴레이션 작성 기술 · 릴레이션의 특징 · 메모리 관리 기법의 이해 · 무결성 제약조건의 이해 · 버전관리 시스템 구축 · 설계 모델링 기술 · 소프트웨어 버전관리 도구 활용 능력 · 오픈소스기반 DBMS 설치 기술 · 운영체제 설치 및 제거 방법 · 운영체제 오류 해결방법 검색 기술 · 운영체제 환경변수 설정 기술 · 운영체제 환경변수 설정 방법 · 인터넷 구성의 개념 · 인터넷 환경변수 설정 기술 · 작업우선순위 변경 기술 · 테이블 선언 및 조작어 · 테이블 제작 및 관리 언어 활용 능력 · 트랜스포트 방식 개념 · 프로세스 스케줄링 기법의 이해 · 프로토콜 개념 · 필요 관련 정보 검색 기술 · 필요 응용 소프트웨어에 대한 이해 · 필요 프로그램 관련 정보 검색 기술 · 사용자의 프로그램 실행 환경에 대한 이해
사전직무경험	<ul style="list-style-type: none"> · 해당사항없음

직무숙련기간	· 약 1년
--------	--------

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020217_16v4
		능 력 단 위	애플리케이션 리팩토링
직무 목적	소스코드가 수행하는 기능을 유지하면서 코드 가독성 및 잠재적 결함을 제거하도록 코드의 구조를 개선할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
리팩토링 기준 수립하기	<ul style="list-style-type: none"> ○ 리팩토링을 적용하여 달성하려는 품질 목표를 수립한다. ○ 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립한다. ○ 리팩토링에 활용할 수 있는 도구 및 절차를 수립한다.
소스 코드 분석하기	<ul style="list-style-type: none"> ○ 개발 표준을 만족하지 않는 소스 코드를 식별한다. ○ 성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별한다. ○ 잠재 결함 요소를 내포한 소드 코드를 식별한다.
리팩토링 적용하기	<ul style="list-style-type: none"> ○ 리팩토링을 위한 소스 코드 구조를 개선한다. ○ 개선한 소스 코드가 초기 기능을 유지하는지 확인한다. ○ 개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토한다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 4년제 대학 졸업 (전공: 전산 관련학)
	· 애플리케이션 리팩토링 실무 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> · IDE 도구 활용 능력 · 국제 코딩 표준 · 디자인 패턴 · 리팩토링 기법 · 리팩토링 적용 기법 · 리팩토링 도구 활용 능력 · 비기능적 품질 특성 · 비기능적 품질 특성 충족 기법 · 설계 패턴 적용 기술

	<ul style="list-style-type: none"> · 역공학 도구 활용 기술 · 좋은 설계 (Good Design) 기준 · 체크리스트 작성 기술 · 코드 동적 분석 도구 활용 기술 · 코드 스멜 유형 · 코드 인스펙션 기술 · 코드 정적 분석 도구 활용 기술 · 클린 코드 특징 · 클린 코드의 필요성 및 특징 · 테스트 도구 활용 기술 · 프로그램 디버깅 능력 · 회귀 테스트
사전직무경험	<ul style="list-style-type: none"> · 데이터입출력 구현 · 요구사항 확인 · 통합구현 · 정보시스템 이행 · 제품소프트웨어 패키징 · 개발자 테스트 · 화면구현 · 인터페이스구현 · 프로그래밍 언어활용 · 응용sw기초기술활용
직무숙련기간	<ul style="list-style-type: none"> · 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020218_16v4
		능 력 단 위	인터페이스 설계
직무 목적	응용소프트웨어 개발을 위해 정의된 시스템 인터페이스 요구사항을 확인하고 인터페이스 대상을 식별하여 인터페이스 설계서를 작성할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
인터페이스 요구사항 확인하기	<ul style="list-style-type: none"> 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 비 기능 요구사항을 분석한다. 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 기능 요구사항을 분석한다. 개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의 정확성과 완전성을 확인한다.
인터페이스 대상 식별하기	<ul style="list-style-type: none"> 개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대상 시스템을 식별한다. 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 연계 시스템을 식별한다. 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 송수신 데이터를 식별한다.
인터페이스 상세 설계하기	<ul style="list-style-type: none"> 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위한 송수신 방법을 명세화한다. 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 데이터를 명세화한다. 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오류 시 처리방안을 명세화한다. 소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성한다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 4년제 대학 졸업	(전공: 전산 관련학)
	· 분산 환경 프로그래밍 과정, 네트워크 프로그래밍 과정	(분야: IT서비스 IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보보안기사 	

	<ul style="list-style-type: none"> · 정보통신기사 · 정보처리산업기사 · 정보보안산업기사 · 정보통신산업기사
지식·기술	<ul style="list-style-type: none"> · IDE 및 개발환경 도구 활용 · UML 작성 기술 · 기술영역별 미들웨어와 솔루션 활용 능력 · 네트워크 · 리뷰 진행 기술 · 산업 분야별 업무특성 · 설계 모델링 기술 · 설계 모델링 방법론 · 소프트웨어 아키텍처 · 시스템 아키텍처 · 요구공학 · 요구사항 관리 도구 사용 능력 · 인터페이스 시스템 · 프레임워크 활용 · 산업 분야별 업무특성
사전직무경험	<ul style="list-style-type: none"> · 데이터입출력 구현 · 요구사항 확인 · 통합구현 · 정보시스템 이행 · 제품소프트웨어 패키징 · 개발자 테스트 · 화면구현 · 인터페이스구현 · 프로그래밍 언어활용 · 응용SW기초기술활용
직무숙련기간	<ul style="list-style-type: none"> · 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020219_16v4
		능 력 단 위	애플리케이션 요구사항 분석
직무 목적	애플리케이션 요구사항 분석이란 구현하고자 하는 애플리케이션의 요구사항을 도출, 분석, 명세화 및 요구사항 검증을 수행할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
요구사항 도출하기	<ul style="list-style-type: none"> ○ 비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구사항 도출을 위한 정보를 수집 할 수 있다. ○ 인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다. ○ 요구사항 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.
요구사항 분석하기	<ul style="list-style-type: none"> ○ 요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구사항과 비기능적 요구사항을 분류 할 수 있다. ○ 도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다. ○ 도출한 요구사항이 개발하고자 하는 애플리케이션 전체 또는 단위 컴포넌트에 향에 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다. ○ 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.
요구사항 명세화하기	<ul style="list-style-type: none"> ○ 애플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의서를 작성할 수 있다. ○ 구현하고자 하는 애플리케이션이 구동되는 시스템이나 하드웨어 관련 내용을 구체화하여 시스템 요구사항 문서를 작성할 수 있다. ○ 구현하고자 하는 애플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)를 작성할 수 있다.
요구사항 검증하기	<ul style="list-style-type: none"> ○ 소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다. ○ 작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한 설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다. ○ 기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다. ○ 분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 4년제 대학 졸업	(전공: 전산 관련학, 경영정보 관련학)
	· SW기획 과정	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사 · 정보기술프로젝트관리전문가 	
지식·기술	<ul style="list-style-type: none"> · SW 제품 개발 수명 주기 이론 · 요구 공학(Requirement Engineering) 이론 · SW 제품 요구사항 관리 절차 · SW 제품 요구사항 도출 지식 · SW 제품 요구사항 정의 기준 · 요구사항 명세서(Software Requirement Specification) 작성 기준 · IEEE830 (IEEE Recommended Practice for Software Requirements pecifications) · 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 · BABOK (Business Analysis Body of Knowledge) · SWEBOK(Software Engineering Body of Knowledge) · SW제품 개발 요구 공학 · SW제품 품질 속성 항목 · SW제품 품질 평가 항목 · 정보시스템의 기술적 지식 · 요구사항 적용 범위 확인 지식 · 요구사항 우선 순위 결정 기준 · 요구사항 영향범위 설정 기준 · 요구사항 도출 기법 · 인터뷰 기법 · 시나리오 작성 기법 · 프로토타입 제작 기법 · 그룹 회의 토의 진행 기법 · 사용자 스토리 작성 기법 · 요구사항 기록 방법 · 요구사항 분석 능력 · 요구사항 분류 기술 · 요구사항의 우선순위를 식별하는 능력 · 사용자와 협상 능력 · 상위레벨 SW제품 아키텍처 수립 기술 	
사전직무경험	<ul style="list-style-type: none"> · 애플리케이션 설계 · 소프트웨어공학 활용 · 애플리케이션 리팩토링 · 인터페이스 설계 	
직무숙련기간	· 약 2년	

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020220_16v1
		능 력 단 위	기능모델 설계
직무 목적	애플리케이션 요구사항에 의해 도출, 분석된 내용을 애플리케이션의 기능으로 명세화하여 요구사항을 검증하고 소프트웨어 개발범위를 확정할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
기능모델 작성하기	<ul style="list-style-type: none"> ○ 기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다. ○ 기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별하여 다이어그램을 작성할 수 있다. ○ 기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능모델 설계 명세서를 작성할 수 있다.
기능모델 검증하기	<ul style="list-style-type: none"> ○ 도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다. ○ 작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다. ○ 작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함하고 있는지 검증할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업
	<ul style="list-style-type: none"> • 전공: 전산 관련학, 소프트웨어공학 관련학) • 전산학, 소프트웨어공학 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> • 컴퓨터시스템응용기술사 • 정보관리기술사 • 정보처리기사 • 정보보안기사
지식·기술	<ul style="list-style-type: none"> • SWEBOK(Software Engineering Body of Knowledge) • 개발 방법론 이해 • 모델 검증 기술 • 모델링 기술 • 모델링 명세화 기술 • 업무 특성에 대한 이해 • 요구 공학(Requirement Engineering) 이론

	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • UML 작성 기술
사전직무경험	<ul style="list-style-type: none"> • 요구사항 확인 • 애플리케이션 설계 • 화면구현 • 데이터 입출력 구현 • 통합구현 • 개발자 테스트 • 정보시스템 이행 • 제품소프트웨어 패키징 • 서버프로그램 구현 • 인터페이스 구현 • 애플리케이션 테스트 • 화면 설계 • 애플리케이션 테스트 관리
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020221_16v4
		능 력 단 위	애플리케이션 설계
직무 목적	요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통모듈 설계, 타 시스템 연동에 대하여 상세 설계할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무가내	책임 및 역할
공통 모듈 설계하기	<ul style="list-style-type: none"> ○ 재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다. ○ 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다. ○ 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다. ○ 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.
타 시스템 연동설계하기	<ul style="list-style-type: none"> ○ 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다. ○ 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다. ○ 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어/솔루션에 대하여 명세를 작성할 수 있다. ○ 소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용	
학습경험	· 4년제 대학 졸업	(전공: 전산 관련학, 소프트웨어 공학 관련학)
	· SW설계 과정, SW개발 과정	(분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보관리기술사 · 컴퓨터시스템응용기술사 · 정보처리기사 · 정보처리산업기사 	

지식·기술	<ul style="list-style-type: none"> • SW 제품 개발 수명 주기 이론 • 요구 공학(Requirement Engineering) 이론 • SW 제품 요구사항 관리 절차 • SW 제품 요구사항 도출 지식 • SW 제품 요구사항 정의 기준 • 요구사항 명세서(Software Requirement Specification) 작성 기준 • IEEE830 (IEEE Recommended Practice for Software Requirements pecifications) • 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 • BABOK (Business Analysis Body of Knowledge) • SWEBOK(Software Engineering Body of Knowledge) • SW제품 개발 요구 공학 • SW제품 품질 속성 항목 • SW제품 품질 평가 항목 • 정보시스템의 기술적 지식 • 요구사항 적용 범위 확인 지식 • 요구사항 우선 순위 결정 기준 • 요구사항 영향범위 설정 기준 • 요구사항 도출 기법 • 인터뷰 기법 • 시나리오 작성 기법 • 프로토타입 제작 기법 • 그룹 회의 토의 진행 기법 • 사용자 스토리 작성 기법 • 요구사항 기록 방법 • 요구사항 분석 능력 • 요구사항 분류 기술 • 요구사항의 우선순위를 식별하는 능력 • 사용자와 협상 능력 • 상위레벨 SW제품 아키텍처 수립 기술 • 품질 속성 Trade-off 방법 • SW제품 요구사항 변경 및 추적 기술 • 비기능 요구사항 식별 기술 • 요구사항 명세 기술 • 수집된 자료에 대한 체계적인 분류 및 정리 기술 • 요구사항 정의서 문서 작성 능력 • 이해당사자와 커뮤니케이션 능력 • 요구사항 검증 기법 • 요구사항 명세서 조사 기법 • 요구사항 명세서 검토 기법 • 요구사항 검증 기준

사전직무경험	<ul style="list-style-type: none">• 애플리케이션 설계• 소프트웨어공학 활용• 애플리케이션 리팩토링• 인터페이스 설계
직무숙련기간	<ul style="list-style-type: none">• 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020222_16v4
		능 력 단 위	정적모델 설계
직무 목적	요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 정적모델을 설계하고 검증할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무가내	책임 및 역할
정적 분석모델 검증하기	<ul style="list-style-type: none"> ○ 소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내역을 확인할 수 있다. ○ 분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.
정적모델 상세화하기	<ul style="list-style-type: none"> ○ 분석된 정적모델을 정적 설계모델로 변환하기 위해 변환규칙을 정의할 수 있다. ○ 정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다. ○ 상세화된 정적 설계모델의 명세서를 작성할 수 있다. ○ 설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업
	<ul style="list-style-type: none"> • SW설계 과정, SW개발 과정
자 격 증	<ul style="list-style-type: none"> • 정보관리기술사 • 컴퓨터시스템응용기술사 • 정보처리기사 • 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 소프트웨어 아키텍처에 대한 이해 • 설계 모델링 기법 • 데이터베이스 이해 • 네트워크 이해 • UML 작성 기술 • 모델링 기술 • 모델 검증 기술E-R 모델 작성 기술 • IDE 및 개발환경 도구 활용 기술

	<ul style="list-style-type: none"> • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 기술 • 기술영역별 미들웨어/솔루션 활용 기술
사전직무경험	<ul style="list-style-type: none"> • 애플리케이션 설계 • 소프트웨어공학 활용 • 애플리케이션 리팩토링 • 인터페이스 설계
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020223_16v4
		능 력 단 위	동적모델 설계
직무 목적	요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 동적모델을 설계하고 검증할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무가내	책임 및 역할
동적 분석모델 검증하기	<ul style="list-style-type: none"> ○ 소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다. ○ 분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.
동적모델 상세화 하기	<ul style="list-style-type: none"> ○ 검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다. ○ 동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다. ○ 상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다. ○ 도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업
	<ul style="list-style-type: none"> • SW설계 과정, SW개발 과정
자 격 증	<ul style="list-style-type: none"> • 정보관리기술사 • 컴퓨터시스템응용기술사 • 정보처리기사 • 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 소프트웨어 아키텍처에 대한 이해 • 설계 모델링 기법 • 데이터베이스 이해 • 네트워크 이해 • UML 작성 기술 • 모델링 기술 • 모델 검증 기술

	<ul style="list-style-type: none"> • E-R 모델 작성 기술 • IDE 및 개발환경 도구 활용 기술 • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 기술 • 기술영역별 미들웨어/솔루션 활용 기술
사전직무경험	<ul style="list-style-type: none"> • 애플리케이션 설계 • 소프트웨어공학 활용 • 애플리케이션 리팩토링 • 인터페이스 설계
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020224_16v4
		능 력 단 위	화면 설계
직무 목적	요구사항분석 단계에서 파악된 화면에 대한 요구사항을 소프트웨어 아키텍처 단계에서 정의된 구현 지침 및 UI/UX 엔지니어가 제시한 UI표준과 지침에 따라 화면을 설계할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무가내	책임 및 역할
UI 요구사항 확인하기	<ul style="list-style-type: none"> ○ 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다. ○ 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다. ○ 작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.
UI 설계하기	<ul style="list-style-type: none"> ○ UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼의 흐름을 설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다. ○ UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다. ○ UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 4년제 대학 졸업 (전공: 전산 관련학, 웹디자인 관련학)
	· UI디자인 과정, SW개발 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 웹디자인기능사 · 정보처리기사 · 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> · 구현에 요구되는 아키텍처의 이해 · 유용성 개념 및 적용원리 · UI 기획/설계/구현 절차 · UI 표준 및 지침 · 유즈케이스에 대한 이해 · 데이터 타입 특성

	<ul style="list-style-type: none"> • 프로토타입 작성법 • 사용자 기반 메뉴 구조 작성 방법 • 감성공학에 대한 이해 • 유즈케이스 작성 능력 • 스토리보드 작성 기술 • 프로토타입 도구 사용 능력 • IDE 도구 활용 능력 • UI 작성 그래픽 도구 활용 능력 • UI 설계도구 활용 능력 • UI 네비게이션 작성 도구 활용 능력 • 화면구성 능력
사전직무경험	<ul style="list-style-type: none"> • 요구사항 확인 • 데이터 입출력 구현 • 통합 구현 • 정보시스템 이행 • 제품소프트웨어 패키 • 서버프로그램 구현 • 애플리케이션 테스트 관리
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020225_16v4
		능 력 단 위	화면 구현
직무 목적	UI 요구사항을 확인하여 설계한 UI 설계를 기반으로 화면을 구현할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무가내	책임 및 역할
UI 설계 확인하기	<ul style="list-style-type: none"> ○ 설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다. ○ UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다. ○ 구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.
UI 구현하기	<ul style="list-style-type: none"> ○ 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다. ○ 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다. ○ 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다. ○ 구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 2년제 대학 졸업 (전공: 전산 관련학, 웹디자인 관련학)
	· UI디자인 과정, SW개발 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 웹디자인기능사 · 정보처리기사 · 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> · 구현에 요구되는 아키텍처의 이해 · 유용성 개념 및 적용원리 · UI 기획/설계/구현 절차 · UI 표준 및 지침 · 유즈케이스에 대한 이해 · 데이터 타입 특성 · 프로토타입 작성법

	<ul style="list-style-type: none"> • 사용자 기반 메뉴 구조 작성 방법 • 감성공학에 대한 이해 • HTML 개념과 적용원리 • CSS의 개념과 적용원리 • 자바스크립트의 개념과 적용원리 • 설계 산출물의 이해 • 프로토타입 도구 사용 능력 • IDE 도구 활용 능력 • UI 작성 그래픽 도구 활용 능력 • UI 설계도구 활용 능력 • UI 네비게이션 작성 도구 활용 능력 • 화면구성 능력 • UI 작성 그래픽 도구 활용 능력 • UI Framework(jQuery, AngularJS등) 활용
사전직무경험	<ul style="list-style-type: none"> • 애플리케이션 배포 • 프로그래밍 언어 활용 • 응용 SW 기초 기술 활용 • 애플리케이션 테스트 수행
직무숙련기간	<ul style="list-style-type: none"> • 약 1년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020226_16v4
		능 력 단 위	애플리케이션 테스트 관리
직무 목적	요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선할 수 있다.		
개발 날짜	22016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
애플리케이션 테스트 케이스 설계하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다. ○ 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다. ○ 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비 할 수 있다.
애플리케이션 개발자 통합 테스트하기	<ul style="list-style-type: none"> ○ 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. ○ 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다. ○ 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	· 2년제 대학 졸업 (전공: 전산 관련학)
	· SW테스트 과정, SW품질 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> · 정보처리기사 · 정보처리산업기사
지식·기술	<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 · 결함관리

	<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 • 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법 • APM(Application Performance Management) 기술
사전직무경험	<ul style="list-style-type: none"> • 요구사항 확인 • 데이터 입출력 구현 • 통합 구현 • 정보시스템 이행 • 제품소프트웨어 패키 • 서버프로그램 구현 • 애플리케이션 테스트 관리
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020227_16v4
		능 력 단 위	애플리케이션 테스트 수행
직무 목적	요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트 케이스에 따라 테스트를 수행하고 결함을 조치할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
애플리케이션 테스트 수행하기	<ul style="list-style-type: none"> ○ 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. ○ 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다. ○ 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.
애플리케이션 개발자 통합 테스트하기	<ul style="list-style-type: none"> ○ 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다. ○ 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다. ○ 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업 (전공: 전산 관련학)
	<ul style="list-style-type: none"> • SW테스트 과정, SW품질 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> • 정보처리기사 • 정보처리산업기사
지식·기술	<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 • 결함관리 • 컴파일러 사용법 • IDE 도구 활용 능력

	<ul style="list-style-type: none"> • 프로그램 디버깅 기법 • 프로그램 코드 검토 기법 • 형상관리 도구 활용 기법 • 테스트 자동화 도구 활용기법 • APM(Application Performance Management) 기술
사전직무경험	<ul style="list-style-type: none"> • 애플리케이션 배포 • 프로그래밍 언어 활용 • 응용 SW 기초 기술 활용 • 화면 구현
직무숙련기간	<ul style="list-style-type: none"> • 약 1년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020228_16v4
		능 력 단 위	소프트웨어공학 활용
직무 목적	응용 소프트웨어 개발과 프로세스 적용활동의 관련된 지식을 소프트웨어의 완전성을 보장하고, 소프트웨어 품질을 평가하기 위해 CASE 도구와 형상관리를 통해 소프트웨어 공학 기술을 적용할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
CASE 도구 활용하기	<ul style="list-style-type: none"> ○ 개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다. ○ CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다. ○ CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.
품질요구사항 확인하기	<ul style="list-style-type: none"> ○ 요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다. ○ 요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다. ○ 개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업 (전공: 전산 관련학, <u>소프트웨어 공학</u> 관련학)
	<ul style="list-style-type: none"> • 소프트웨어 공학 과정 (분야: IT서비스, IT기술)
자 격 증	<ul style="list-style-type: none"> • 정보관리기술사 • 컴퓨터시스템응용기술사 • 정보처리기사 • 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> • 소프트웨어 개발 방법론 • 프로젝트 관리 방법론 • 소프트웨어 개발 표준 • 요구공학 방법론 • 비용산정 모델

	<ul style="list-style-type: none"> • 아키텍처 평가 • 소프트웨어 요구사항 명세 기준에 따른 품질 평가에 대한 지식 • 국제 표준 제품 품질특성에 대한 사항 • 품질 점검 기법(리뷰, 워크스루, 인스펙션 등) • 품질 관리 프로세스 • 시정 조치 관리 프로세스 • 감사 프로세스 • 소프트웨어 개발 프레임워크에 대한 지식 • 협상 기술 • 의사결정 기술 • 프로세스 테일러링 기술 • UML 작성기술 • 모델링 기법 • 유즈케이스 모델 정의할 수 있는 기술 • 형상관리 기법 • 리뷰 진행 기술 • 인스펙션 진행 기술 • 감사 기술 • 요구사항명세서 작성기술 • 소프트웨어 요구사항 품질 측정, 점검 기술
사전직무경험	<ul style="list-style-type: none"> • 소프트웨어공학 활용 • 애플리케이션 리팩토링 • 기능모델 설계 • 동적모델 설계 • SW개발방법론 활용
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

□ 직무 기본 정보

직 무	응용SW엔지니어링	능력단위분류번호	2001020229_16v4
		능 력 단 위	SW개발방법론 활용
직무 목적	응용소프트웨어 특성에 따라 정형화된 개발방법론을 선정하고, 프로젝트 특성에 맞도록 테일러링 하여 응용소프트웨어 개발에 활용할 수 있다.		
개발 날짜	2016. 06. 30	개 발 기 관	한국소프트웨어산업협회

□ 직무 책임 및 역할

주 요 업 무	책임 및 역할
SW 개발방법론 선정하기	<ul style="list-style-type: none"> ○ 개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된 개발방법론 선정기준으로 활용할 수 있다. ○ 정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.
SW개발방법론 테일러링하기	<ul style="list-style-type: none"> ○ 프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다. ○ 확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다. ○ 용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다. ○ 테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다.

□ 직무수행 요건

구 분	상 세 내 용
학습경험	<ul style="list-style-type: none"> • 4년제 대학 졸업
	<ul style="list-style-type: none"> • 소프트웨어 공학 과정
자 격 증	<ul style="list-style-type: none"> • 정보관리기술사 • 컴퓨터시스템응용기술사 • 정보처리기사 • 정보처리산업기사
지식·기술	<ul style="list-style-type: none"> • 소프트웨어 생명주기 모델 • 소프트웨어 개발 방법론 • 요구공학 방법론 • 비용산정 모델 • 반복, 점증 프로세스 • 프로젝트 관리 방법론 • 소프트웨어 개발 표준 • 소프트웨어 개발 프레임워크에 대한 지식

	<ul style="list-style-type: none"> • 모델링 기법 • 의사결정 기술 • 모델링 도구 활용기술 • 협상 기술 • 의사결정 기술 • 감사 기술 • 프로세스 테일러링 기술 • UML 작성기술 • 형상관리 기법
사전직무경험	<ul style="list-style-type: none"> • 소프트웨어공학 활용 • 애플리케이션 리팩토링 • 기능모델 설계 • 동적모델 설계 • SW개발방법론 활용
직무숙련기간	<ul style="list-style-type: none"> • 약 2년

3.1. 채용·배치·승진체크리스트 개요

- 개념 : 근로자를 채용하거나 배치하거나 승진시키기 위하여 각 개인이 해당 직급에서 요구되는 직업능력을 어느 정도 가지고 있는지 확인하기 위한 진단도구
- 구성요소 : ① 목적, ② 직급명, ③ 인적사항, ④ 능력구분, ⑤ 평가영역, ⑥ 평가문항, ⑦ 답변기재란, ⑧ 평가결과로 구성

【 채용·배치·승진 체크리스트 구성요소 】

구 성 요 소	세 부 내 용
목적	• 평가를 실시하는 방향이나 이유로 채용, 배치, 승진이 있음
직급명	• 해당 조직에서 일의 종류나 난이도, 책임도 등의 유사성을 기준으로 구분한 등급
인적사항	• 평가하고자 하는 예비근로자 및 근로자의 성명, 직위, 성별 등과 같은 개인적 특성
능력구분	• 평가하고자 하는 직급에서 요구되는 직업능력의 구분(직업기초능력, 직무수행능력)
평가영역	• 직업기초능력과 직무수행능력의 하위영역
평가문항	• 예비근로자 및 근로자의 지식이나 활동을 측정하기 위한 측정가능하고 구체적인 문장
답변기재란	• 평가자가 평가문항을 읽고 평가대상자의 행동과 일치하는 정도에 직접 표기하는 부분
평가결과	• 기재한 답변을 합산하여 점수를 산출하고 해석

3.2. 채용·배치·승진체크리스트

목적 : <input type="checkbox"/> 채용 <input type="checkbox"/> 배치 <input type="checkbox"/> 승진	SW 프로그래머
--	----------

이 름 :

직 위 :

성 별 :

특이사항 :

[직업기초능력]

평 가 영 역	평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
의사소통능력	직장생활에서 필요한 문서를 확인하고, 읽고, 내용을 이해하여 업무 수행에 필요한 요점을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 목적과 상황에 적합한 아이디어와 정보를 전달할 수 있는 문서를 작성할 수 있다.	①	②	③	④	⑤
	다른 사람의 말을 주의 깊게 듣고 적절하게 반응할 수 있다.	①	②	③	④	⑤
	목적과 상황에 맞는 말과 비언어적 행동을 통해 아이디어와 정보를 찾고, 이를 효과적으로 전달할 수 있다.	①	②	③	④	⑤
	외국어로 된 간단한 자료를 이해하거나, 간단한 외국인의 의사표현을 이해하고, 자신의 업무와 관련하여 필요한 기초외국어능력을 기를 수 있다.	①	②	③	④	⑤
수리능력	직장생활에서 필요한 기초적인 사칙연산과 계산방법을 이해하고 활용하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 평균, 합계, 빈도와 같은 기초적인 통계기법을 활용하여 자료의 특성과 경향성을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)의 의미를 파악하고, 필요한 정보를 해석하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)를 이용하여 결과를 효과적으로 제시하는 능력을 기를 수 있다.	①	②	③	④	⑤
문제해결능력	직장생활에서 발생한 문제를 해결하기 위해서 창의적, 논리적, 비판적으로 생각할 수 있다.	①	②	③	④	⑤
	직장생활에서 발생한 문제를 올바르게 인식하고 적절한 해결책을 적용하여 해결할 수 있다.	①	②	③	④	⑤

자기개발능력	직장생활에서 다양한 방법으로 자신의 장단점, 흥미, 적성 등을 분석하여 자신의 가치를 설명할 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신의 역할과 목표를 정립하고, 이를 위하여 자신의 행동과 업무수행을 관리하고 통제할 수 있다.	①	②	③	④	⑤
	직업인으로서 자신의 경력단계를 이해하고 이에 적절한 경력 개발 계획을 수립할 수 있다.	①	②	③	④	⑤
자원관리능력	직장생활에서 필요한 시간을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 예산을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 물적자원을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 인적자원을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
대인관계능력	직장생활에서 다른 구성원들과 목표를 공유하고 원만한 관계를 유지하며, 자신의 역할을 이해하고 책임감있게 업무를 수행할 수 있다.	①	②	③	④	⑤
	직장생활 중 조직구성원들의 업무향상에 도움을 주며 동기화시킬 수 있고, 조직의 목표 및 비전을 제시할 수 있다.	①	②	③	④	⑤
	직장생활에서 조직구성원 사이에 갈등이 발생하였을 경우 이를 원만히 조절할 수 있다.	①	②	③	④	⑤
	직장생활에서 협상 가능한 목표를 세우고 상황에 맞는 협상 전략을 선택하여 다른 사람과 협상하는 능력을 기를 수 있다.	①	②	③	④	⑤
	고객서비스에 대한 이해를 바탕으로 실제 현장에서 다양한 고객에 대처할 수 있으며, 고객만족을 이끌어 낼 수 있는 능력을 기를 수 있다.	①	②	③	④	⑤
정보능력	직장생활에서 컴퓨터 관련이론을 이해하여 업무수행을 위해 인터넷과 소프트웨어를 활용할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 정보를 찾아내고, 업무수행에 적합하게 조직 · 관리하여 활용할 수 있다.	①	②	③	④	⑤
기술능력	기본적인 직장생활에 필요한 기술의 원리 및 절차를 이해하는 능력을 기를 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 선택할 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 실제로 적용하고 결과를 확인할 수 있다.	①	②	③	④	⑤

조직이해능력	직장생활에서 직업인으로서 다른 나라의 문화를 이해하고 국제적인 동향을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 구조와 목적, 문화, 규칙 등과 같은 조직체제를 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 경영목표와 경영방법을 이해하고, 경영의 한 주체로서 조직경영에 참여하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신에게 주어진 업무의 성격과 내용을 알고 업무처리절차에 따라 효과적으로 업무를 수행할 수 있다.	①	②	③	④	⑤
직업윤리	직업윤리를 실천하기 위하여 근면하고 정직하며 성실하게 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤
	직업윤리를 실천하기 위하여 봉사하며, 책임있고, 규칙을 준수하고, 예의바른 태도로 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤

[직무수행능력]

평 가 영 역		평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 배포	애플리케이션 배포 환경 구성하기	결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다.	①	②	③	④	⑤
		애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다.	①	②	③	④	⑤
		애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다.	①	②	③	④	⑤
		설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.	①	②	③	④	⑤
	애플리케이션 소스 검증하기	정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다.	①	②	③	④	⑤
		소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다.	①	②	③	④	⑤
		소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.	①	②	③	④	⑤
	애플리케이션 빌드하기	작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다.	①	②	③	④	⑤
		애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다.	①	②	③	④	⑤
		애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다.	①	②	③	④	⑤
		애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되	①	②	③	④	⑤

	애플리케이션 배포하기	었는지 여부를 확인할 수 있다.	①	②	③	④	⑤
		애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명할 수 있다.	①	②	③	④	⑤
		애플리케이션 실행 환경에 대한 정보를 확인할 수 있다.	①	②	③	④	⑤
		애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다.	①	②	③	④	⑤
		애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다.	①	②	③	④	⑤
프로그래밍 언어 활용	기본문법 활용하기	애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다.	①	②	③	④	⑤
		프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.	①	②	③	④	⑤
	언어특성 활용하기	프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.	①	②	③	④	⑤
		파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.	①	②	③	④	⑤
		프로그래밍 언어별 특성을 파악하고 설명할 수 있다.	①	②	③	④	⑤
	라이브러리 활용하기	애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용 할 수 있다.	①	②	③	④	⑤
		애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다.	①	②	③	④	⑤
		애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.	①	②	③	④	⑤
		선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.	①	②	③	④	⑤
응용 SW 기초 기술 활용	운영체제 기초 활용하기	CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다.	①	②	③	④	⑤
		응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다.	①	②	③	④	⑤
		운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다.	①	②	③	④	⑤
	데이터베이스 기초 활용하기	주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.	①	②	③	④	⑤
		데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다.	①	②	③	④	⑤
		데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.	①	②	③	④	⑤
	네트워크 기초 활용하기	네트워크 계층구조에서 각 층의 역할을 설명할 수 있다.	①	②	③	④	⑤
		응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다.	①	②	③	④	⑤
		패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다.	①	②	③	④	⑤

	기본 개발환경 구축하기	응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다.	①	②	③	④	⑤
		응용개발에 필요한 개발도구를 설치하고 운용할 수 있다.	①	②	③	④	⑤
		웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용할 수 있다.	①	②	③	④	⑤
화면 구현	UI 설계 확인하기	설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다.	①	②	③	④	⑤
		구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.	①	②	③	④	⑤
		UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다.	①	②	③	④	⑤
	UI 구현하기	소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다.	①	②	③	④	⑤
		확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다	①	②	③	④	⑤
		확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다.	①	②	③	④	⑤
		구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
애플리케이션 테스트 수행	애플리케이션 테스트 수행하기	애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다.	①	②	③	④	⑤
	애플리케이션 결함 조치하기	애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.	①	②	③	④	⑤
		결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.	①	②	③	④	⑤

[평가결과]

영역	점수
직업기초능력	<u>영역별 점수 합산</u>
직무수행능력	<u>영역별 점수 합산</u>
합계	<u>점수 합계</u>

목적 : <input type="checkbox"/> 채용 <input type="checkbox"/> 배치 <input type="checkbox"/> 승진	(구)(가칭)응용SW 실무개발자
--	-------------------

이 름 :

직 위 :

성 별 :

특이사항 :

[직업기초능력]

평 가 영 역	평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
의사소통능력	직장생활에서 필요한 문서를 확인하고, 읽고, 내용을 이해하여 업무 수행에 필요한 요점을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 목적과 상황에 적합한 아이디어와 정보를 전달할 수 있는 문서를 작성할 수 있다.	①	②	③	④	⑤
	다른 사람의 말을 주의 깊게 듣고 적절하게 반응할 수 있다.	①	②	③	④	⑤
	목적과 상황에 맞는 말과 비언어적 행동을 통해 아이디어와 정보를 찾고, 이를 효과적으로 전달할 수 있다.	①	②	③	④	⑤
	외국어로 된 간단한 자료를 이해하거나, 간단한 외국인의 의사표현을 이해하고, 자신의 업무와 관련하여 필요한 기초외국어능력을 기를 수 있다.	①	②	③	④	⑤
수리능력	직장생활에서 필요한 기초적인 사칙연산과 계산방법을 이해하고 활용하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 평균, 합계, 빈도와 같은 기초적인 통계기법을 활용하여 자료의 특성과 경향성을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)의 의미를 파악하고, 필요한 정보를 해석하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)를 이용하여 결과를 효과적으로 제시하는 능력을 기를 수 있다.	①	②	③	④	⑤
문제해결능력	직장생활에서 발생한 문제를 해결하기 위해서 창의적, 논리적, 비판적으로 생각할 수 있다.	①	②	③	④	⑤
	직장생활에서 발생한 문제를 올바르게 인식하고 적절한 해결책을 적용하여 해결할 수 있다.	①	②	③	④	⑤

자기개발능력	직장생활에서 다양한 방법으로 자신의 장단점, 흥미, 적성 등을 분석하여 자신의 가치를 설명할 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신의 역할과 목표를 정립하고, 이를 위하여 자신의 행동과 업무수행을 관리하고 통제할 수 있다.	①	②	③	④	⑤
	직업인으로서 자신의 경력단계를 이해하고 이에 적절한 경력 개발 계획을 수립할 수 있다.	①	②	③	④	⑤
자원관리능력	직장생활에서 필요한 시간을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 예산을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 물적자원을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 인적자원을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
대인관계능력	직장생활에서 다른 구성원들과 목표를 공유하고 원만한 관계를 유지하며, 자신의 역할을 이해하고 책임감있게 업무를 수행할 수 있다.	①	②	③	④	⑤
	직장생활 중 조직구성원들의 업무향상에 도움을 주며 동기화시킬 수 있고, 조직의 목표 및 비전을 제시할 수 있다.	①	②	③	④	⑤
	직장생활에서 조직구성원 사이에 갈등이 발생하였을 경우 이를 원만히 조절할 수 있다.	①	②	③	④	⑤
	직장생활에서 협상 가능한 목표를 세우고 상황에 맞는 협상 전략을 선택하여 다른 사람과 협상하는 능력을 기를 수 있다.	①	②	③	④	⑤
	고객서비스에 대한 이해를 바탕으로 실제 현장에서 다양한 고객에 대처할 수 있으며, 고객만족을 이끌어 낼 수 있는 능력을 기를 수 있다.	①	②	③	④	⑤
정보능력	직장생활에서 컴퓨터 관련이론을 이해하여 업무수행을 위해 인터넷과 소프트웨어를 활용할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 정보를 찾아내고, 업무수행에 적합하게 조직 · 관리하여 활용할 수 있다.	①	②	③	④	⑤
기술능력	기본적인 직장생활에 필요한 기술의 원리 및 절차를 이해하는 능력을 기를 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 선택할 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 실제로 적용하고 결과를 확인할 수 있다.	①	②	③	④	⑤

조직이해능력	직장생활에서 직업인으로서 다른 나라의 문화를 이해하고 국제적인 동향을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 구조와 목적, 문화, 규칙 등과 같은 조직체제를 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 경영목표와 경영방법을 이해하고, 경영의 한 주체로서 조직경영에 참여하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신에게 주어진 업무의 성격과 내용을 알고 업무처리절차에 따라 효과적으로 업무를 수행할 수 있다.	①	②	③	④	⑤
직업윤리	직업윤리를 실천하기 위하여 근면하고 정직하며 성실하게 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤
	직업윤리를 실천하기 위하여 봉사하며, 책임있고, 규칙을 준수하고, 예의바른 태도로 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤

[직무수행능력]

평 가 영 역		평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 구현(구버전)	개발환경 구축하기	응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.	①	②	③	④	⑤
		사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.	①	②	③	④	⑤
	공통 모듈 구현하기	공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.	①	②	③	④	⑤
		소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다.	①	②	③	④	⑤
		개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
	서버 프로그램 구현하기	애플리케이션 설계를 기반으로, 프로그래밍 언어와 도구를 활용하여 서버 프로그램 구현 기술에 부합하는 서버 프로그램을 개발할 수 있다.	①	②	③	④	⑤
		클라이언트 프로그램에 대한 종속도를 낮출 수 있고 쉽게 연동할 수 있는 서버 프로그램을 개발할 수 있다.	①	②	③	④	⑤
		개발된 서버 프로그램 내부 기능과 제공하는 인터페이스에	①	②	③	④	⑤

		대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.					
	배치 프로그램 구현하기	목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 개발할 수 있다.	①	②	③	④	⑤
		애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 개발할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 배치 프로그램을 개발할 수 있다.	①	②	③	④	⑤
		개발된 배치 프로그램을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
	개발자 단위 테스트하기	구현한 응용소프트웨어 단위가 설계 내용을 반영하는지 여부를 판단하기 위한 단위테스트의 표준, 절차, 기법 등을 정의할 수 있다.	①	②	③	④	⑤
		기능요구사항을 분석하여 단위테스트 계획을 수립하고, 단위테스트 계획대로 단위 모듈/컴포넌트 별로 테스트를 수행할 수 있다.	①	②	③	④	⑤
		단위 모듈/컴포넌트가 설계 내용을 만족하는지 여부를 계획한 단위 테스트 케이스에 따라 검증할 수 있다.	①	②	③	④	⑤
		단위 테스트 결과 발견된 결함과 이슈를 식별하고, 단위 테스트 결과 분석을 통하여 테스트의 충분성 여부를 검증할 수 있다.	①	②	③	④	⑤
		단위 테스트 결과 발견된 결함에 대한 개선의 시스템 반영 여부를 검증하고, 필요할 경우 시정조치를 실시할 수 있다.	①	②	③	④	⑤
	애플리케이션 성능 개선하기	실 데이터를 기반으로 테스트를 수행하여 애플리케이션의 성능을 확인하고 목표 성능이 충족되도록 개선할 수 있다.	①	②	③	④	⑤
		애플리케이션 성능을 개선하기 위해, 프로그래밍 언어와 이의 표준에 대한 이해를 바탕으로 소스코드에 내재된 품질 수준을 분석하기 위한 도구를 활용할 수 있다.	①	②	③	④	⑤
		애플리케이션 성능을 개선하기 위해, 기 정의된 프로그래밍 언어 표준 및 가이드라인에 따른 코드 품질 메트릭을 이해하고 적용할 수 있다.	①	②	③	④	⑤
화면 구현(구버전)	UI 요구사항 확인하기	응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.	①	②	③	④	⑤
		작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.	①	②	③	④	⑤
	UI 설계하기	UI 요구사항과 UI 표준 및 지침에 따라, 화면과 품의 흐름을	①	②	③	④	⑤

		설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다.					
		UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다.	①	②	③	④	⑤
		UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.	①	②	③	④	⑤
	UI 구현하기	화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, UI 요구사항이 반영된 화면과 폼을 구현할 수 있다.	①	②	③	④	⑤
		화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다.	①	②	③	④	⑤
		구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
개발자 테스트(구버전)	개발자 테스트케이스 설계하기	개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다.	①	②	③	④	⑤
		개발자 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비 할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다.	①	②	③	④	⑤
	개발자 통합 테스트하기	개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다.	①	②	③	④	⑤
		개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
		개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.	①	②	③	④	⑤
	개발자 결함 조치하기	결함이 발생한 소스를 분석하여 추가적인 코딩으로 결함을 제거하고, 결함조치 시 기존에 구현된 로직과의 연관성을 고려하여 부작용을 최소화 할 수 있다.	①	②	③	④	⑤
		개발자 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.	①	②	③	④	⑤
		개발자 통합 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.	①	②	③	④	⑤
	애플리케이션 테스트(구버전)	개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트	①	②	③	④	⑤

		시작 및 종료 조건 등을 준비 할 수 있다.					
	애플리케이션 테스트 수행하기	애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.	①	②	③	④	⑤
	애플리케이션 결함 조치하기	애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.	①	②	③	④	⑤
		결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.	①	②	③	④	⑤
	애플리케이션 성능 개선하기	애플리케이션 테스트를 통하여 애플리케이션의 성능을 분석하고, 성능 저하 요인을 발견할 수 있다.	①	②	③	④	⑤
		코드 최적화 기법, 아키텍처 조정 및 호출 순서 조정 등을 적용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤
		프로그래밍 언어의 특성에 대한 이해를 기반으로 소스코드 품질 분석 도구를 활용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤

[평가결과]

영역	점수
직업기초능력	<u>영역별 점수 합산</u>
직무수행능력	<u>영역별 점수 합산</u>
합계	<u>점수 합계</u>

목적 : <input type="checkbox"/> 채용 <input type="checkbox"/> 배치 <input type="checkbox"/> 승진	응용SW 엔지니어
--	-----------

이 름 :

직 위 :

성 별 :

특이사항 :

[직업기초능력]

평 가 영 역	평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
의사소통능력	직장생활에서 필요한 문서를 확인하고, 읽고, 내용을 이해하여 업무 수행에 필요한 요점을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 목적과 상황에 적합한 아이디어와 정보를 전달할 수 있는 문서를 작성할 수 있다.	①	②	③	④	⑤
	다른 사람의 말을 주의 깊게 듣고 적절하게 반응할 수 있다.	①	②	③	④	⑤
	목적과 상황에 맞는 말과 비언어적 행동을 통해 아이디어와 정보를 찾고, 이를 효과적으로 전달할 수 있다.	①	②	③	④	⑤
	외국어로 된 간단한 자료를 이해하거나, 간단한 외국인의 의사표현을 이해하고, 자신의 업무와 관련하여 필요한 기초외국어능력을 기를 수 있다.	①	②	③	④	⑤
수리능력	직장생활에서 필요한 기초적인 사칙연산과 계산방법을 이해하고 활용하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 평균, 합계, 빈도와 같은 기초적인 통계기법을 활용하여 자료의 특성과 경향성을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)의 의미를 파악하고, 필요한 정보를 해석하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)를 이용하여 결과를 효과적으로 제시하는 능력을 기를 수 있다.	①	②	③	④	⑤
문제해결능력	직장생활에서 발생한 문제를 해결하기 위해서 창의적, 논리적, 비판적으로 생각할 수 있다.	①	②	③	④	⑤
	직장생활에서 발생한 문제를 올바르게 인식하고 적절한 해결책을 적용하여 해결할 수 있다.	①	②	③	④	⑤

자기개발능력	직장생활에서 다양한 방법으로 자신의 장단점, 흥미, 적성 등을 분석하여 자신의 가치를 설명할 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신의 역할과 목표를 정립하고, 이를 위하여 자신의 행동과 업무수행을 관리하고 통제할 수 있다.	①	②	③	④	⑤
	직업인으로서 자신의 경력단계를 이해하고 이에 적절한 경력 개발 계획을 수립할 수 있다.	①	②	③	④	⑤
자원관리능력	직장생활에서 필요한 시간을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 예산을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 물적자원을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 인적자원을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
대인관계능력	직장생활에서 다른 구성원들과 목표를 공유하고 원만한 관계를 유지하며, 자신의 역할을 이해하고 책임감있게 업무를 수행할 수 있다.	①	②	③	④	⑤
	직장생활 중 조직구성원들의 업무향상에 도움을 주며 동기화시킬 수 있고, 조직의 목표 및 비전을 제시할 수 있다.	①	②	③	④	⑤
	직장생활에서 조직구성원 사이에 갈등이 발생하였을 경우 이를 원만히 조절할 수 있다.	①	②	③	④	⑤
	직장생활에서 협상 가능한 목표를 세우고 상황에 맞는 협상 전략을 선택하여 다른 사람과 협상하는 능력을 기를 수 있다.	①	②	③	④	⑤
	고객서비스에 대한 이해를 바탕으로 실제 현장에서 다양한 고객에 대처할 수 있으며, 고객만족을 이끌어 낼 수 있는 능력을 기를 수 있다.	①	②	③	④	⑤
정보능력	직장생활에서 컴퓨터 관련이론을 이해하여 업무수행을 위해 인터넷과 소프트웨어를 활용할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 정보를 찾아내고, 업무수행에 적합하게 조직 · 관리하여 활용할 수 있다.	①	②	③	④	⑤
기술능력	기본적인 직장생활에 필요한 기술의 원리 및 절차를 이해하는 능력을 기를 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 선택할 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 실제로 적용하고 결과를 확인할 수 있다.	①	②	③	④	⑤

조직이해능력	직장생활에서 직업인으로서 다른 나라의 문화를 이해하고 국제적인 동향을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 구조와 목적, 문화, 규칙 등과 같은 조직체제를 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 경영목표와 경영방법을 이해하고, 경영의 한 주체로서 조직경영에 참여하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신에게 주어진 업무의 성격과 내용을 알고 업무처리절차에 따라 효과적으로 업무를 수행할 수 있다.	①	②	③	④	⑤
직업윤리	직업윤리를 실천하기 위하여 근면하고 정직하며 성실하게 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤
	직업윤리를 실천하기 위하여 봉사하며, 책임있고, 규칙을 준수하고, 예의바른 태도로 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤

[직무수행능력]

평 가 영 역		평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
요구사항 확인	현행 시스템 분석하기	개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다.	①	②	③	④	⑤
		현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.	①	②	③	④	⑤
	요구사항 확인하기	소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다.	①	②	③	④	⑤
		업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다.	①	②	③	④	⑤
		업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.	①	②	③	④	⑤
	분석모델 확인하기	소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다.	①	②	③	④	⑤
		업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를	①	②	③	④	⑤

		개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.					
		업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다.	①	②	③	④	⑤
애플리케이션 설계(구버전)	정적모델 상세설계하기	소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세설계 내역을 확인하고 검증할 수 있다.	①	②	③	④	⑤
		정적모델 설계를 상세 분석하여 도출된 엔티티 클래스 또는 데이터 엔티티를 상세화 할 수 있다.	①	②	③	④	⑤
		정적모델 설계를 상세 분석하여 제어 클래스 또는 사용자와 시스템 간 상호작용에 관련된 제어관계를 상세화 할 수 있다.	①	②	③	④	⑤
		정적모델 설계를 상세 분석하여 경계 클래스 또는 사용자와 시스템 간의 상호작용 케이스를 상세화 할 수 있다.	①	②	③	④	⑤
		정적모델 설계를 상세 분석하여 클래스 모델 또는 시스템의 정적 구조를 상세화 할 수 있다.	①	②	③	④	⑤
	동적모델 상세설계하기	소프트웨어 아키텍처 설계 가이드라인을 참조하여, 동적모델 상세설계의 내역 확인 및 검증할 수 있다.	①	②	③	④	⑤
		동적모델 설계를 상세 분석하여, 응용소프트웨어의 구현을 위해 협동 다이어그램을 통해 유즈케이스를 상세화 할 수 있다.	①	②	③	④	⑤
		동적모델 설계를 상세 분석하여, 응용소프트웨어의 구현을 위해 시퀀스 다이어그램을 작성할 수 있다.	①	②	③	④	⑤
		동적모델 설계를 상세 분석하여, 응용소프트웨어의 구현을 위한 상태차트를 작성할 수 있다.	①	②	③	④	⑤
	공통 모듈 설계하기	재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다.	①	②	③	④	⑤
		개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다.	①	②	③	④	⑤
		소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다.	①	②	③	④	⑤
		전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.	①	②	③	④	⑤
	타 시스템 연동설계하기	소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사	①	②	③	④	⑤

		용될 기술영역별 미들웨어/솔루션에 대하여 명세를 작성할 수 있다.					
데이터 입출력 구현	논리 데이터저장소 확인하기	논리 데이터저장소 설계 내역에서 데이터의 논리적 단위와 데이터 간의 관계를 확인할 수 있다.	①	②	③	④	⑤
		업무 분석가, 데이터베이스 엔지니어가 작성한 논리 데이터 저장소 설계 내역에서 정의된 데이터의 유형을 확인하고 식별할 수 있다.	①	②	③	④	⑤
		논리 데이터저장소 설계 내역에서 데이터 또는 데이터간의 제약조건과 이들 간의 관계를 식별할 수 있다.	①	②	③	④	⑤
	물리 데이터저장소 설계하기	논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을 반영한 물리 데이터저장소 설계를 수행할 수 있다.	①	②	③	④	⑤
		물리 데이터저장소 설계에 따라 데이터저장소에 실제 데이터가 저장될 물리적 공간을 구성할 수 있다.	①	②	③	④	⑤
		논리 데이터저장소 설계를 바탕으로 목표 시스템의 데이터 특성을 반영하여 최적화된 물리 데이터저장소를 설계할 수 있다.	①	②	③	④	⑤
	데이터 조작 프로시저 작성하기	응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소로부터 데이터를 읽어 오는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소에 연결을 수행하는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 변경 내용 또는 신규 입력된 데이터를 데이터 저장소에 저장하는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
		구현된 데이터 조작 프로시저를 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
	데이터 조작 프로시저 최적화하기	프로그래밍 언어와 도구에 대한 이해를 바탕으로 응용소프트웨어 설계, 물리 데이터저장소 설계와 운영 환경을 고려하여 데이터 조작 프로시저의 성능을 예측할 수 있다.	①	②	③	④	⑤
		업무 분석가에 의해 정의된 요구사항을 기준으로, 성능측정 도구를 활용하여 데이터 조작 프로시저의 성능을 측정할 수 있다.	①	②	③	④	⑤
		테스트 결과와 정의된 요구사항을 기준으로 데이터조작 프로시저의 성능에 따른 이슈 발생 시 이에 대해 해결할 수 있다.	①	②	③	④	⑤
		실 데이터를 기반으로 테스트를 수행하여 데이터 조작 프로시저의 성능에 영향을 주는 병목을 파악할 수 있다.	①	②	③	④	⑤
통합 구현	연계 데이터 구성하기	개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다.	①	②	③	④	⑤

		둘 간의 데이터 연계 요구사항을 분석할 수 있다.					
		개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.	①	②	③	④	⑤
	연계 매커니즘 구성하기	개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다.	①	②	③	④	⑤
		개발하고자하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다.	①	②	③	④	⑤
		응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.	①	②	③	④	⑤
	내외부 연계 모듈 구현하기	구성된 연계 매커니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다.	①	②	③	④	⑤
		구성된 연계 매커니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다.	①	②	③	④	⑤
		연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다.	①	②	③	④	⑤
		구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
정보시스템 이행	정보시스템 사용자 교육하기	개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다.	①	②	③	④	⑤
		개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다.	①	②	③	④	⑤
		교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육결과 및 성과를 평가할 수 있다.	①	②	③	④	⑤
		효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검 할 수 있다.	①	②	③	④	⑤
	정보시스템 이행하기	하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다.	①	②	③	④	⑤
		시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발할 수 있다.	①	②	③	④	⑤
		정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증할 수 있다.	①	②	③	④	⑤
		정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.	①	②	③	④	⑤
	정보시스템 안정화하기	정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다.	①	②	③	④	⑤
		정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정	①	②	③	④	⑤

		보시시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다.					
		개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.	①	②	③	④	⑤
제품소프트웨어 패키징	제품소프트웨어 패키징하기	이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다.	①	②	③	④	⑤
		저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.	①	②	③	④	⑤
		신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다.	①	②	③	④	⑤
		사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.	①	②	③	④	⑤
	제품소프트웨어 매뉴얼 작성하기	사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.	①	②	③	④	⑤
		사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.	①	②	③	④	⑤
		형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다.	①	②	③	④	⑤
	제품소프트웨어 버전관리하기	형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다.	①	②	③	④	⑤
		버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.	①	②	③	④	⑤
		생명주기와 개발 방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다.	①	②	③	④	⑤
소프트웨어공학 활용(구버전)	소프트웨어 개발방법론 활용하기	응용소프트웨어의 특성에 따른 정형화된 개발 방법론을 선정하고 응용소프트웨어 개발에 개발 방법론을 적용할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법을 정립할 수 있다.	①	②	③	④	⑤
		CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다.	①	②	③	④	⑤
	CASE 도구 활용하기	개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다.	①	②	③	④	⑤
		CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.	①	②	③	④	⑤
		요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다.	①	②	③	④	⑤
	품질요구사항 확인하기		①	②	③	④	⑤

		요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다.	①	②	③	④	⑤
		개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용 소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.	①	②	③	④	⑤
서버프로그램 구현	개발환경 구축하기	응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다.	①	②	③	④	⑤
		사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.	①	②	③	④	⑤
	공통 모듈 구현하기	공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.	①	②	③	④	⑤
		소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다.	①	②	③	④	⑤
		개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
	서버 프로그램 구현하기	개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다.	①	②	③	④	⑤
		세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다.	①	②	③	④	⑤
		업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다.	①	②	③	④	⑤
		개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.	①	②	③	④	⑤
	배치 프로그램 구현하기	애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.	①	②	③	④	⑤
		목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.	①	②	③	④	⑤
		개발된 배치 프로그램을 테스트를 수행할 수 있다.	①	②	③	④	⑤
인터페이스 구현	인터페이스 설계서 확인하기	인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.	①	②	③	④	⑤

화면 설계	인터페이스 기능 구현하기	개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다.	①	②	③	④	⑤
		응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다.	①	②	③	④	⑤
	인터페이스 구현 검증하기	구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다.	①	②	③	④	⑤
		인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다.	①	②	③	④	⑤
		인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.	①	②	③	④	⑤
	UI 요구사항 확인하기	응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.	①	②	③	④	⑤
		작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다.	①	②	③	④	⑤
	UI 설계하기	UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다.	①	②	③	④	⑤
		UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼 흐름을 설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다.	①	②	③	④	⑤
		UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.	①	②	③	④	⑤
애플리케이션 테스트 관리	애플리케이션 테스트케이스 설계하기	개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비할 수 있다.	①	②	③	④	⑤
	애플리케이션 통합 테스트하기	개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
		개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다.	①	②	③	④	⑤
		개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분	①	②	③	④	⑤

		성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.					
	애플리케이션 성능 개선하기	코드 최적화 기법, 아키텍처 조정 및 호출 순서 조정 등을 적용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤
		애플리케이션 테스트를 통하여 애플리케이션의 성능을 분석하고, 성능 저하 요인을 발견할 수 있다.	①	②	③	④	⑤
		프로그래밍 언어의 특성에 대한 이해를 기반으로 소스코드 품질 분석 도구를 활용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤

[평가결과]

영역	점수
직업기초능력	<u>영역별 점수 합산</u>
직무수행능력	<u>영역별 점수 합산</u>
합계	<u>점수 합계</u>

목적 : <input type="checkbox"/> 채용 <input type="checkbox"/> 배치 <input type="checkbox"/> 승진	응용SW 분석설계자
--	------------

이 름 :

직 위 :

성 별 :

특이사항 :

[직업기초능력]

평 가 영 역	평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
의사소통능력	직장생활에서 필요한 문서를 확인하고, 읽고, 내용을 이해하여 업무 수행에 필요한 요점을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 목적과 상황에 적합한 아이디어와 정보를 전달할 수 있는 문서를 작성할 수 있다.	①	②	③	④	⑤
	다른 사람의 말을 주의 깊게 듣고 적절하게 반응할 수 있다.	①	②	③	④	⑤
	목적과 상황에 맞는 말과 비언어적 행동을 통해 아이디어와 정보를 찾고, 이를 효과적으로 전달할 수 있다.	①	②	③	④	⑤
	외국어로 된 간단한 자료를 이해하거나, 간단한 외국인의 의사표현을 이해하고, 자신의 업무와 관련하여 필요한 기초외국어능력을 기를 수 있다.	①	②	③	④	⑤
수리능력	직장생활에서 필요한 기초적인 사칙연산과 계산방법을 이해하고 활용하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 평균, 합계, 빈도와 같은 기초적인 통계기법을 활용하여 자료의 특성과 경향성을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)의 의미를 파악하고, 필요한 정보를 해석하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)를 이용하여 결과를 효과적으로 제시하는 능력을 기를 수 있다.	①	②	③	④	⑤
문제해결능력	직장생활에서 발생한 문제를 해결하기 위해서 창의적, 논리적, 비판적으로 생각할 수 있다.	①	②	③	④	⑤
	직장생활에서 발생한 문제를 올바르게 인식하고 적절한 해결책을 적용하여 해결할 수 있다.	①	②	③	④	⑤

자기개발능력	직장생활에서 다양한 방법으로 자신의 장단점, 흥미, 적성 등을 분석하여 자신의 가치를 설명할 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신의 역할과 목표를 정립하고, 이를 위하여 자신의 행동과 업무수행을 관리하고 통제할 수 있다.	①	②	③	④	⑤
	직업인으로서 자신의 경력단계를 이해하고 이에 적절한 경력 개발 계획을 수립할 수 있다.	①	②	③	④	⑤
자원관리능력	직장생활에서 필요한 시간을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 예산을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 물적자원을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 인적자원을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
대인관계능력	직장생활에서 다른 구성원들과 목표를 공유하고 원만한 관계를 유지하며, 자신의 역할을 이해하고 책임감있게 업무를 수행할 수 있다.	①	②	③	④	⑤
	직장생활 중 조직구성원들의 업무향상에 도움을 주며 동기화시킬 수 있고, 조직의 목표 및 비전을 제시할 수 있다.	①	②	③	④	⑤
	직장생활에서 조직구성원 사이에 갈등이 발생하였을 경우 이를 원만히 조절할 수 있다.	①	②	③	④	⑤
	직장생활에서 협상 가능한 목표를 세우고 상황에 맞는 협상 전략을 선택하여 다른 사람과 협상하는 능력을 기를 수 있다.	①	②	③	④	⑤
	고객서비스에 대한 이해를 바탕으로 실제 현장에서 다양한 고객에 대처할 수 있으며, 고객만족을 이끌어 낼 수 있는 능력을 기를 수 있다.	①	②	③	④	⑤
정보능력	직장생활에서 컴퓨터 관련이론을 이해하여 업무수행을 위해 인터넷과 소프트웨어를 활용할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 정보를 찾아내고, 업무수행에 적합하게 조직 · 관리하여 활용할 수 있다.	①	②	③	④	⑤
기술능력	기본적인 직장생활에 필요한 기술의 원리 및 절차를 이해하는 능력을 기를 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 선택할 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 실제로 적용하고 결과를 확인할 수 있다.	①	②	③	④	⑤

조직이해능력	직장생활에서 직업인으로서 다른 나라의 문화를 이해하고 국제적인 동향을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 구조와 목적, 문화, 규칙 등과 같은 조직체제를 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 경영목표와 경영방법을 이해하고, 경영의 한 주체로서 조직경영에 참여하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신에게 주어진 업무의 성격과 내용을 알고 업무처리절차에 따라 효과적으로 업무를 수행할 수 있다.	①	②	③	④	⑤
직업윤리	직업윤리를 실천하기 위하여 근면하고 정직하며 성실하게 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤
	직업윤리를 실천하기 위하여 봉사하며, 책임있고, 규칙을 준수하고, 예의바른 태도로 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤

[직무수행능력]

평 가 영 역		평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 리팩토링	리팩토링 기준 수립하기	리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다.	①	②	③	④	⑤
		리팩토링에 활용할 수 있는 도구 및 절차를 수립할 수 있다.	①	②	③	④	⑤
		리팩토링을 적용하여 달성하려는 품질 목표를 수립할 수 있다.	①	②	③	④	⑤
	소스 코드 분석하기	성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별할 수 있다.	①	②	③	④	⑤
		개발 표준을 만족하지 않는 소스 코드를 식별할 수 있다.	①	②	③	④	⑤
		잠재 결함 요소를 내포한 소스 코드를 식별할 수 있다.	①	②	③	④	⑤
	리팩토링 적용하기	리팩토링을 위한 소스 코드 구조를 개선할 수 있다.	①	②	③	④	⑤
		개선한 소스 코드가 초기 기능을 유지하는지 확인할 수 있다.	①	②	③	④	⑤
		개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토할 수 있다.	①	②	③	④	⑤
인터페이스 설계	인터페이스 요구사항 확인하기	정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 비 기능 요구사항을 분석할 수 있다.	①	②	③	④	⑤
		정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 기능 요구사항을 분석할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의	①	②	③	④	⑤

		정확성과 완전성을 확인한다.					
	인터페이스 대상 식별하기	개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대상 시스템을 식별할 수 있다.	①	②	③	④	⑤
		개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 송수신 데이터를 식별할 수 있다.	①	②	③	④	⑤
		개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 연계 시스템을 식별할 수 있다.	①	②	③	④	⑤
	인터페이스 상세 설계하기	개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위한 송수신 방법을 명세화할 수 있다.	①	②	③	④	⑤
		개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 데이터를 명세화할 수 있다.	①	②	③	④	⑤
		개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오류 시 처리방안을 명세화할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따 라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성할 수 있다.	①	②	③	④	⑤
기능 모델링	기능 모델 작성하기	기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다.	①	②	③	④	⑤
		기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능 모델 설계 명세서를 작성할 수 있다.	①	②	③	④	⑤
		기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별 하여 다이어그램을 작성할 수 있다.	①	②	③	④	⑤
	기능 모델 검증하기	도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다.	①	②	③	④	⑤
		작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다.	①	②	③	④	⑤
		작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함 하고 있는지 검증할 수 있다.	①	②	③	④	⑤
애플리케이션 설계	공통 모듈 설계하기	소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모 듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있 다.	①	②	③	④	⑤
		개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다.	①	②	③	④	⑤
		재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다.	①	②	③	④	⑤
		전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도 록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계 할 수 있다.	①	②	③	④	⑤
	타 시스템 연동설계하기	소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드 라인을 작성할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이	①	②	③	④	⑤

		드라인에 따라, 타 시스템 연동 상세 설계할 수 있다.					
		소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어솔루션에 대하여 명세를 작성할 수 있다.	①	②	③	④	⑤
		소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	①	②	③	④	⑤
정적모델 설계	정적 분석모델 검증하기	소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내역을 확인할 수 있다.	①	②	③	④	⑤
		분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	①	②	③	④	⑤
	정적모델 상세화하기	분석된 정적모델을 정적 설계모델로 변환하기 위해 변환규칙을 정의할 수 있다.	①	②	③	④	⑤
		정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다.	①	②	③	④	⑤
		상세화된 정적 설계모델의 명세서를 작성할 수 있다.	①	②	③	④	⑤
		설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.	①	②	③	④	⑤
동적모델 설계	동적 분석모델 검증하기	소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다.	①	②	③	④	⑤
		분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	①	②	③	④	⑤
	동적모델 상세화하기	검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다.	①	②	③	④	⑤
		동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다.	①	②	③	④	⑤
		도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.	①	②	③	④	⑤
		상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다.	①	②	③	④	⑤
소프트웨 어공학 활용	CASE 도구 활용하기	CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다.	①	②	③	④	⑤
		개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다.	①	②	③	④	⑤
		CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.	①	②	③	④	⑤
	품질요구사항 확인하기	요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다.	①	②	③	④	⑤
		요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다.	①	②	③	④	⑤
		개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용 소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.	①	②	③	④	⑤
소프트웨	소프트웨어개발	개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된	①	②	③	④	⑤

어개발 방법론 활용	방법론 선정하기	개발방법론 선정기준으로 활용할 수 있다.					
		정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.	①	②	③	④	⑤
	소프트웨어개발 방법론 테일러링하기	프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다.	①	②	③	④	⑤
		응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다.	①	②	③	④	⑤
		확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다.	①	②	③	④	⑤
		테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다	①	②	③	④	⑤

[평가결과]

영역	점수
직업기초능력	<u>영역별 점수 합산</u>
직무수행능력	<u>영역별 점수 합산</u>
합계	<u>점수 합계</u>

목적 : <input type="checkbox"/> 채용 <input type="checkbox"/> 배치 <input type="checkbox"/> 승진	요구사항 분석가
--	----------

이 름 :
 직 위 :
 성 별 :
 특이사항 :

[직업기초능력]

평 가 영 역	평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
의사소통능력	직장생활에서 필요한 문서를 확인하고, 읽고, 내용을 이해하여 업무 수행에 필요한 요점을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 목적과 상황에 적합한 아이디어와 정보를 전달할 수 있는 문서를 작성할 수 있다.	①	②	③	④	⑤
	다른 사람의 말을 주의 깊게 듣고 적절하게 반응할 수 있다.	①	②	③	④	⑤
	목적과 상황에 맞는 말과 비언어적 행동을 통해 아이디어와 정보를 찾고, 이를 효과적으로 전달할 수 있다.	①	②	③	④	⑤
	외국어로 된 간단한 자료를 이해하거나, 간단한 외국인의 의사표현을 이해하고, 자신의 업무와 관련하여 필요한 기초외국어능력을 기를 수 있다.	①	②	③	④	⑤
수리능력	직장생활에서 필요한 기초적인 사칙연산과 계산방법을 이해하고 활용하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 평균, 합계, 빈도와 같은 기초적인 통계기법을 활용하여 자료의 특성과 경향성을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)의 의미를 파악하고, 필요한 정보를 해석하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 도표(그림, 표, 그래프 등)를 이용하여 결과를 효과적으로 제시하는 능력을 기를 수 있다.	①	②	③	④	⑤
문제해결능력	직장생활에서 발생한 문제를 해결하기 위해서 창의적, 논리적, 비판적으로 생각할 수 있다.	①	②	③	④	⑤
	직장생활에서 발생한 문제를 올바르게 인식하고 적절한 해결책을 적용하여 해결할 수 있다.	①	②	③	④	⑤

자기개발능력	직장생활에서 다양한 방법으로 자신의 장단점, 흥미, 적성 등을 분석하여 자신의 가치를 설명할 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신의 역할과 목표를 정립하고, 이를 위하여 자신의 행동과 업무수행을 관리하고 통제할 수 있다.	①	②	③	④	⑤
	직업인으로서 자신의 경력단계를 이해하고 이에 적절한 경력 개발 계획을 수립할 수 있다.	①	②	③	④	⑤
자원관리능력	직장생활에서 필요한 시간을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 예산을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 물적자원을 확인하고, 확보하여 업무 수행에 이를 할당할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 인적자원을 확인하고, 확보하여 업무 수행에 이를 할당하는 능력을 기를 수 있다.	①	②	③	④	⑤
대인관계능력	직장생활에서 다른 구성원들과 목표를 공유하고 원만한 관계를 유지하며, 자신의 역할을 이해하고 책임감있게 업무를 수행할 수 있다.	①	②	③	④	⑤
	직장생활 중 조직구성원들의 업무향상에 도움을 주며 동기화시킬 수 있고, 조직의 목표 및 비전을 제시할 수 있다.	①	②	③	④	⑤
	직장생활에서 조직구성원 사이에 갈등이 발생하였을 경우 이를 원만히 조절할 수 있다.	①	②	③	④	⑤
	직장생활에서 협상 가능한 목표를 세우고 상황에 맞는 협상 전략을 선택하여 다른 사람과 협상하는 능력을 기를 수 있다.	①	②	③	④	⑤
	고객서비스에 대한 이해를 바탕으로 실제 현장에서 다양한 고객에 대처할 수 있으며, 고객만족을 이끌어 낼 수 있는 능력을 기를 수 있다.	①	②	③	④	⑤
정보능력	직장생활에서 컴퓨터 관련이론을 이해하여 업무수행을 위해 인터넷과 소프트웨어를 활용할 수 있다.	①	②	③	④	⑤
	직장생활에서 필요한 정보를 찾아내고, 업무수행에 적합하게 조직 · 관리하여 활용할 수 있다.	①	②	③	④	⑤
기술능력	기본적인 직장생활에 필요한 기술의 원리 및 절차를 이해하는 능력을 기를 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 선택할 수 있다.	①	②	③	④	⑤
	기본적인 직장생활에 필요한 기술을 실제로 적용하고 결과를 확인할 수 있다.	①	②	③	④	⑤

조직이해능력	직장생활에서 직업인으로서 다른 나라의 문화를 이해하고 국제적인 동향을 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 구조와 목적, 문화, 규칙 등과 같은 조직체제를 파악하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신이 속한 조직의 경영목표와 경영방법을 이해하고, 경영의 한 주체로서 조직경영에 참여하는 능력을 기를 수 있다.	①	②	③	④	⑤
	직장생활에서 직업인으로서 자신에게 주어진 업무의 성격과 내용을 알고 업무처리절차에 따라 효과적으로 업무를 수행할 수 있다.	①	②	③	④	⑤
직업윤리	직업윤리를 실천하기 위하여 근면하고 정직하며 성실하게 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤
	직업윤리를 실천하기 위하여 봉사하며, 책임있고, 규칙을 준수하고, 예의바른 태도로 업무에 임하는 자세를 배양할 수 있다.	①	②	③	④	⑤

[직무수행능력]

평 가 영 역		평 가 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 요구사항 분석	요구사항 도출하기	요구사항 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.	①	②	③	④	⑤
		인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다.	①	②	③	④	⑤
		비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구사항 도출을 위한 정보를 수집 할 수 있다.	①	②	③	④	⑤
	요구사항 분석하기	도출한 요구사항이 개발하고자 하는 어플리케이션 전체 또는 단위 컴포넌트에 향해 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다.	①	②	③	④	⑤
		도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다.	①	②	③	④	⑤
		요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구사항과 비기능적 요구사항을 분류 할 수 있다.	①	②	③	④	⑤
		요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.	①	②	③	④	⑤
	요구사항 명세화하기	애플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의를 작성할 수 있다.	①	②	③	④	⑤
		구현하고자 하는 어플리케이션이 구동되는 시스템이나 하드	①	②	③	④	⑤

		웨어 관련 내용을 구체화 하여 시스템 요구사항 문서를 작성할 수 있다.					
		구현하고자 하는 애플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)을 작성할 수 있다.	①	②	③	④	⑤
	요구사항 검증하기	소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다.	①	②	③	④	⑤
		작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한 설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다.	①	②	③	④	⑤
		기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다.	①	②	③	④	⑤
		분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.	①	②	③	④	⑤

[평가결과]

영역	점수
직업기초능력	<u>영역별 점수 합산</u>
직무수행능력	<u>영역별 점수 합산</u>
합계	<u>점수 합계</u>

4.1. 자가진단도구 개요

- 개념 : 업무를 성공적으로 수행하는데 요구되는 능력과 근로자 자신의 보유 능력을 비교·점
점해 볼 수 있는 도구
- 구성요소 : ① 번호체계, ② 진단항목, ③ 지시문, ④ 진단영역, ⑤ 진단문항, ⑥ 답변기재
란, ⑦ 진단결과로 구성

【 자가진단도구의 구성요소 】

구 성 요 소	세 부 내 용
번호체계	• 직업능력 자가진단도구를 분류하기 위하여 직업능력별로 부여된 번호
진단항목	• 진단하고자 하는 직업능력명
지시문	• 진단문항을 읽고 답변을 기재하는 방법에 대한 안내문
진단영역	• 진단하고자 하는 직업능력을 구성하는 하위영역과 세부영역
진단문항	• 근로자(응답자)의 지식이나 활동을 측정하기 위한 측정가능하고 구체적인 문장
답변기재란	• 근로자(응답자)가 진단문항을 읽고 자신의 상황이나 생각과 일치하는 정도에 직접 표기 하는 부분
진단결과	• 기재한 답변을 합산하여 점수를 산출하고 해석

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
현행 시스템 분석하기	1. 나는 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.	①	②	③	④	⑤
요구사항 확인하기	1. 나는 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다.	①	②	③	④	⑤
	3. 나는 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.	①	②	③	④	⑤
분석모델 확인하기	1. 나는 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다.	①	②	③	④	⑤
	3. 나는 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
현행 시스템 분석하기	3		
요구사항 확인하기	3		
분석모델 확인하기	3		
합계	9		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
논리 데이터저장 소 확인하기	1. 나는 업무 분석가, 데이터베이스 엔지니어가 작성한 논리 데이터저장소 설계 내역에서 정의된 데이터의 유형을 확인하고 식별할 수 있다.	①	②	③	④	⑤
	2. 나는 논리 데이터저장소 설계 내역에서 데이터의 논리적 단위와 데이터 간의 관계를 확인할 수 있다.	①	②	③	④	⑤
	3. 나는 논리 데이터저장소 설계 내역에서 데이터 또는 데이터간의 제약 조건과 이들 간의 관계를 식별할 수 있다.	①	②	③	④	⑤
물리 데이터저장 소 설계하기	1. 나는 논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을 반영한 물리 데이터저장소 설계를 수행할 수 있다.	①	②	③	④	⑤
	2. 나는 논리 데이터저장소 설계를 바탕으로 목표 시스템의 데이터 특성을 반영하여 최적화된 물리 데이터저장소를 설계할 수 있다.	①	②	③	④	⑤
	3. 나는 물리 데이터저장소 설계에 따라 데이터저장소에 실제 데이터가 저장될 물리적 공간을 구성할 수 있다.	①	②	③	④	⑤
데이터 조작 프로시저 작성하기	1. 나는 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소에 연결을 수행하는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 저장소로부터 데이터를 읽어 오는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 응용소프트웨어 설계와 물리 데이터저장소 설계에 따라 데이터 변경 내용 또는 신규 입력된 데이터를 데이터 저장소에 저장하는 프로시저를 작성할 수 있다.	①	②	③	④	⑤
	4. 나는 구현된 데이터 조작 프로시저를 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화할 수 있다.	①	②	③	④	⑤
데이터 조작 프로시저 최적화하기	1. 나는 프로그래밍 언어와 도구에 대한 이해를 바탕으로 응용소프트웨어 설계, 물리 데이터저장소 설계와 운영 환경을 고려하여 데이터 조작 프로시저의 성능을 예측할 수 있다.	①	②	③	④	⑤
	2. 나는 업무 분석가에 의해 정의된 요구사항을 기준으로, 성능측정 도구를 활용하여 데이터 조작 프로시저의 성능을 측정할 수 있다.	①	②	③	④	⑤
	3. 나는 실 데이터를 기반으로 테스트를 수행하여 데이터 조작 프로시저의 성능에 영향을 주는 병목을 파악할 수 있다.	①	②	③	④	⑤
	4. 나는 테스트 결과와 정의된 요구사항을 기준으로 데이터조작 프로시저의 성능에 따른 이슈 발생 시 이에 대해 해결할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
논리 데이터저장소 확인하기	3		
물리 데이터저장소 설계하기	3		
데이터 조작 프로시저 작성하기	4		
데이터 조작 프로시저 최적화하기	4		
합계	14		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
연계 데이터 구성하기	1. 나는 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.	①	②	③	④	⑤
연계 메카니즘 구성하기	1. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다.	①	②	③	④	⑤
	4. 나는 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.	①	②	③	④	⑤
내외부 연계 모듈 구현하기	1. 나는 구성된 연계 메카니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다.	①	②	③	④	⑤
	2. 나는 구성된 연계 메카니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다.	①	②	③	④	⑤
	3. 나는 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다.	①	②	③	④	⑤
	4. 나는 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
연계 데이터 구성하기	3		
연계 메카니즘 구성하기	4		
내외부 연계 모듈 구현하기	4		
합계	11		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
정보시스템 사용자 교육하기	1. 나는 개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다.	①	②	③	④	⑤
	2. 나는 개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다.	①	②	③	④	⑤
	3. 나는 효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검 할 수 있다.	①	②	③	④	⑤
	4. 나는 교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육 결과 및 성과를 평가할 수 있다.	①	②	③	④	⑤
정보시스템 이행하기	1. 나는 하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다.	①	②	③	④	⑤
	2. 나는 시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발할 수 있다.	①	②	③	④	⑤
	3. 나는 정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증할 수 있다.	①	②	③	④	⑤
	4. 나는 정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.	①	②	③	④	⑤
정보시스템 안정화하기	1. 나는 정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다.	①	②	③	④	⑤
	2. 나는 정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정보시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다.	①	②	③	④	⑤
	3. 나는 개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
정보시스템 사용자 교육하기	4		
정보시스템 이행하기	4		
정보시스템 안정화하기	3		
합계	11		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
제품소프트웨어 패키징하기	1. 나는 신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다.	①	②	③	④	⑤
	2. 나는 이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.	①	②	③	④	⑤
제품소프트웨어 매뉴얼 작성하기	1. 나는 사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품 소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품 소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품 소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.	①	②	③	④	⑤
제품소프트웨어 버전관리하기	1. 나는 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다.	①	②	③	④	⑤
	2. 나는 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다.	①	②	③	④	⑤
	3. 나는 버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
제품소프트웨어 패키징하기	3		
제품소프트웨어 매뉴얼 작성하기	3		
제품소프트웨어 버전관리하기	3		
합계	9		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요하다.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
개발환경 구축하기	1. 나는 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다.	①	②	③	④	⑤
	2. 나는 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다.	①	②	③	④	⑤
	3. 나는 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상 관리 환경을 구축할 수 있다.	①	②	③	④	⑤
공통 모듈 구현하기	1. 나는 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다.	①	②	③	④	⑤
	3. 나는 개발된 공통 모듈의 내부 기능과 제공하는 인터페이스에 대해 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤
서버 프로그램 구현하기	1. 나는 업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다.	①	②	③	④	⑤
	4. 나는 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.	①	②	③	④	⑤
배치 프로그램 구현하기	1. 나는 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.	①	②	③	④	⑤
	2. 나는 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.	①	②	③	④	⑤
	3. 나는 개발된 배치 프로그램을 테스트를 수행할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
개발환경 구축하기	3		
공통 모듈 구현하기	3		
서버 프로그램 구현하기	4		

배치 프로그램 구현하기	3		
합계	13		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
인터페이스 설계서 확인하기	1. 나는 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.	①	②	③	④	⑤
인터페이스 기능 구현하기	1. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다.	①	②	③	④	⑤
	4. 나는 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.	①	②	③	④	⑤
인터페이스 구현 검증하기	1. 나는 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다.	①	②	③	④	⑤
	2. 나는 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다.	①	②	③	④	⑤
	3. 나는 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
인터페이스 설계서 확인하기	3		
인터페이스 기능 구현하기	4		
인터페이스 구현 검증하기	3		
합계	10		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 배포 환경 구성하기	1. 나는 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다.	①	②	③	④	⑤
	2. 나는 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다.	①	②	③	④	⑤
	3. 나는 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다.	①	②	③	④	⑤
	4. 나는 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.	①	②	③	④	⑤
애플리케이션 소스 검증하기	1. 나는 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다.	①	②	③	④	⑤
	2. 나는 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다.	①	②	③	④	⑤
	3. 나는 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.	①	②	③	④	⑤
애플리케이션 빌드하기	1. 나는 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스 코드를 빌드 시스템으로 이관할 수 있다.	①	②	③	④	⑤
	2. 나는 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다.	①	②	③	④	⑤
	4. 나는 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지 여부를 확인할 수 있다.	①	②	③	④	⑤
	5. 나는 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명할 수 있다.	①	②	③	④	⑤
애플리케이션 배포하기	1. 나는 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다.	①	②	③	④	⑤
	3. 나는 애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다.	①	②	③	④	⑤
	4. 나는 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
애플리케이션 배포 환경 구성하기	4		
애플리케이션 소스 검증하기	3		
애플리케이션 빌드하기	5		

애플리케이션 배포하기	4		
합계	16		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
기본문법 활용하기	1. 나는 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다.	①	②	③	④	⑤
	2. 나는 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다.	①	②	③	④	⑤
	3. 나는 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.	①	②	③	④	⑤
언어특성 활용하기	1. 나는 프로그래밍 언어별 특성을 파악하고 설명할 수 있다.	①	②	③	④	⑤
	2. 나는 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다.	①	②	③	④	⑤
	3. 나는 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용할 수 있다.	①	②	③	④	⑤
라이브러리 활용하기	1. 나는 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다.	①	②	③	④	⑤
	2. 나는 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성할 수 있다.	①	②	③	④	⑤
	3. 나는 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
기본문법 활용하기	3		
언어특성 활용하기	3		
라이브러리 활용하기	3		
합계	9		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
운영체제 기초 활용하기	1. 나는 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다.	①	②	③	④	⑤
	2. 나는 CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다.	①	②	③	④	⑤
	3. 나는 운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다.	①	②	③	④	⑤
데이터베이스 기초 활용하기	1. 나는 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다.	①	②	③	④	⑤
	2. 나는 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다.	①	②	③	④	⑤
	3. 나는 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.	①	②	③	④	⑤
네트워크 기초 활용하기	1. 나는 네트워크 계층구조에서 각 층의 역할을 설명할 수 있다.	①	②	③	④	⑤
	2. 나는 응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다.	①	②	③	④	⑤
	3. 나는 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다.	①	②	③	④	⑤
기본 개발환경 구축하기	1. 나는 응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다.	①	②	③	④	⑤
	2. 나는 응용개발에 필요한 개발도구를 설치하고 운용할 수 있다.	①	②	③	④	⑤
	3. 나는 웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
운영체제 기초 활용하기	3		
데이터베이스 기초 활용하기	3		
네트워크 기초 활용하기	3		
기본 개발환경 구축하기	3		
합계	12		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
리팩토링 기준 수립하기	1. 나는 리팩토링을 적용하여 달성하려는 품질 목표를 수립할 수 있다.	①	②	③	④	⑤
	2. 나는 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다.	①	②	③	④	⑤
	3. 나는 리팩토링에 활용할 수 있는 도구 및 절차를 수립할 수 있다.	①	②	③	④	⑤
소스 코드 분석하기	1. 나는 개발 표준을 만족하지 않는 소스 코드를 식별할 수 있다.	①	②	③	④	⑤
	2. 나는 성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 잠재 결함 요소를 내포한 소드 코드를 식별할 수 있다.	①	②	③	④	⑤
리팩토링 적용하기	1. 나는 리팩토링을 위한 소스 코드 구조를 개선할 수 있다.	①	②	③	④	⑤
	2. 나는 개선한 소스 코드가 초기 기능을 유지하는지 확인할 수 있다.	①	②	③	④	⑤
	3. 나는 개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
리팩토링 기준 수립하기	3		
소스 코드 분석하기	3		
리팩토링 적용하기	3		
합계	9		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
인터페이스 요구사항 확인하기	1. 나는 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 비 기능 요구사항을 분석할 수 있다.	①	②	③	④	⑤
	2. 나는 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 기능 요구사항을 분석할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의 정확 성과 완전성을 확인한다.	①	②	③	④	⑤
인터페이스 대상 식별하기	1. 나는 개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대 상 시스템을 식별할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위 해 필요한 연계 시스템을 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위 해 필요한 송수신 데이터를 식별할 수 있다.	①	②	③	④	⑤
인터페이스 상세 설계하기	1. 나는 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위 한 송수신 방법을 명세화할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위 해 필요한 데이터를 명세화할 수 있다.	①	②	③	④	⑤
	3. 나는 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오 류 시 처리방안을 명세화할 수 있다.	①	②	③	④	⑤
	4. 나는 소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
인터페이스 요구사항 확인하기	3		
인터페이스 대상 식별하기	3		
인터페이스 상세 설계하기	4		
합계	10		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
요구사항 도출하기	1. 나는 비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구 사항 도출을 위한 정보를 수집 할 수 있다.	①	②	③	④	⑤
	2. 나는 인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다.	①	②	③	④	⑤
	3. 나는 요구사항 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.	①	②	③	④	⑤
요구사항 분석하기	1. 나는 요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구 사항과 비기능적 요구사항을 분류 할 수 있다.	①	②	③	④	⑤
	2. 나는 도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다.	①	②	③	④	⑤
	3. 나는 도출한 요구사항이 개발하고자 하는 어플리케이션 전체 또는 단위 컴포넌트에 향해 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다.	①	②	③	④	⑤
	4. 나는 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.	①	②	③	④	⑤
요구사항 명세화하기	1. 나는 어플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의서를 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 구현하고자 하는 어플리케이션이 구동되는 시스템이나 하드웨어 관련 내용을 구체화 하여 시스템 요구사항 문서를 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 구현하고자 하는 어플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)을 작성할 수 있다.	①	②	③	④	⑤
요구사항 검증하기	1. 나는 소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다.	①	②	③	④	⑤
	2. 나는 작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한 설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다.	①	②	③	④	⑤
	3. 나는 기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다.	①	②	③	④	⑤
	4. 나는 분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
요구사항 도출하기	3		
요구사항 분석하기	4		
요구사항 명세화하기	3		
요구사항 검증하기	4		
합계	14		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
기능 모델 작성하기	1. 나는 기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다.	①	②	③	④	⑤
	2. 나는 기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별하여 다이어그램을 작성할 수 있다.	①	②	③	④	⑤
	3. 나는 기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능모델 설계 명세서를 작성할 수 있다.	①	②	③	④	⑤
기능 모델 검증하기	1. 나는 도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다.	①	②	③	④	⑤
	2. 나는 작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다.	①	②	③	④	⑤
	3. 나는 작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함하고 있는지 검증할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
기능 모델 작성하기	3		
기능 모델 검증하기	3		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
공통 모듈 설계하기	1. 나는 재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다.	①	②	③	④	⑤
	3. 나는 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다.	①	②	③	④	⑤
	4. 나는 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.	①	②	③	④	⑤
타 시스템 연동설계하 기	1. 나는 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다.	①	②	③	④	⑤
	2. 나는 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다.	①	②	③	④	⑤
	3. 나는 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어솔루션에 대하여 명세를 작성할 수 있다.	①	②	③	④	⑤
	4. 나는 소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
공통 모듈 설계하기	4		
타 시스템 연동설계하기	4		
합계	8		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
정적 분석모델 검증하기	1. 나는 소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내역을 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	①	②	③	④	⑤
정적모델 상세화하기	1. 나는 분석된 정적모델을 정적 설계모델로 변환하기 위해 변환규칙을 정의할 수 있다.	①	②	③	④	⑤
	2. 나는 정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다.	①	②	③	④	⑤
	3. 나는 상세화된 정적 설계모델의 명세서를 작성할 수 있다.	①	②	③	④	⑤
	4. 나는 설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
정적 분석모델 검증하기	2		
정적모델 상세화하기	4		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
동적 분석모델 검증하기	1. 나는 소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	①	②	③	④	⑤
동적모델 상세화하기	1. 나는 검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다.	①	②	③	④	⑤
	2. 나는 동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다.	①	②	③	④	⑤
	3. 나는 상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다.	①	②	③	④	⑤
	4. 나는 도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
동적 분석모델 검증하기	2		
동적모델 상세화하기	4		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
UI 요구사항 확인하기	1. 나는 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다.	①	②	③	④	⑤
	2. 나는 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다.	①	②	③	④	⑤
	3. 나는 작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.	①	②	③	④	⑤
UI 설계하기	1. 나는 UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼의 흐름을 설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다.	①	②	③	④	⑤
	2. 나는 UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다.	①	②	③	④	⑤
	3. 나는 UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
UI 요구사항 확인하기	3		
UI 설계하기	3		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
UI 설계 확인하기	1. 나는 설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다.	①	②	③	④	⑤
	2. 나는 UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다.	①	②	③	④	⑤
	3. 나는 구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.	①	②	③	④	⑤
UI 구현하기	1. 나는 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다.	①	②	③	④	⑤
	2. 나는 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다	①	②	③	④	⑤
	3. 나는 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다.	①	②	③	④	⑤
	4. 나는 구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
UI 설계 확인하기	3		
UI 구현하기	4		
합계	7		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 테스트케이스 설계하기	1. 나는 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다.	①	②	③	④	⑤
	2. 나는 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다.	①	②	③	④	⑤
	3. 나는 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비 할 수 있다.	①	②	③	④	⑤
애플리케이션 통합 테스트하기	1. 나는 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
	2. 나는 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다.	①	②	③	④	⑤
	3. 나는 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.	①	②	③	④	⑤
애플리케이션 성능 개선하기	1. 나는 애플리케이션 테스트를 통하여 애플리케이션의 성능을 분석하고, 성능 저하 요인을 발견할 수 있다.	①	②	③	④	⑤
	2. 나는 코드 최적화 기법, 아키텍처 조정 및 호출 순서 조정 등을 적용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤
	3. 나는 프로그래밍 언어의 특성에 대한 이해를 기반으로 소스코드 품질 분석 도구를 활용하여 애플리케이션 성능을 개선할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
애플리케이션 테스트케이스 설계하기	3		
애플리케이션 통합 테스트하기	3		
애플리케이션 성능 개선하기	3		
합계	9		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
애플리케이션 테스트 수행하기	1. 나는 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다.	①	②	③	④	⑤
	2. 나는 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록할 수 있다.	①	②	③	④	⑤
	3. 나는 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.	①	②	③	④	⑤
애플리케이션 결함 조치하기	1. 나는 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.	①	②	③	④	⑤
	2. 나는 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다.	①	②	③	④	⑤
	3. 나는 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
애플리케이션 테스트 수행하기	3		
애플리케이션 결함 조치하기	3		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
CASE 도구 활용하기	1. 나는 개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하 는 최적의 CASE 도구를 선정할 수 있다.	①	②	③	④	⑤
	2. 나는 CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다.	①	②	③	④	⑤
	3. 나는 CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.	①	②	③	④	⑤
품질요구사항 확인하기	1. 나는 요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정 의하고 품질 평가항목과 지침을 제공할 수 있다.	①	②	③	④	⑤
	2. 나는 요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바 르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다.	①	②	③	④	⑤
	3. 나는 개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용소프 트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
CASE 도구 활용하기	3		
품질요구사항 확인하기	3		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련
이나 개인학습을 통한 개발이 필요함.

진단영역	진 단 문 항	매우 미흡	미흡	보통	우수	매우 우수
소프트웨어개발 방법론 선정하기	1. 나는 개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된 개발방법론 선정기준으로 활용할 수 있다.	①	②	③	④	⑤
	2. 나는 정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.	①	②	③	④	⑤
소프트웨어개발 방법론 테일러링하기	1. 나는 프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다.	①	②	③	④	⑤
	2. 나는 확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다.	①	②	③	④	⑤
	3. 나는 응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다.	①	②	③	④	⑤
	4. 나는 테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다	①	②	③	④	⑤

[진단결과]

진단영역	문항 수	점수	점수 ÷ 문항 수
소프트웨어개발 방법론 선정하기	2		
소프트웨어개발 방법론 테일러링하기	4		
합계	6		

☞ 자신의 점수를 문항 수로 나눈 값이 '3점'이하에 해당하는 영역은 업무를 성공적으로 수행하는데 요구는 능력이 부족한 것으로 교육훈련이나 개인학습을 통한 개발이 필요함.

2 훈 련 기 준

□ 개발목적

- 체계적이고 효과적인 직업능력개발을 위하여 훈련의 대상이 되는 직종별로 훈련의 목표, 교과내용 및 시설·장비와 교사 등에 관한 훈련기준 개발(근로자 직업능력개발법 제38조)

* 내용구성 : 훈련의 목표, 교과목 및 그 내용, 시설 및 장비, 훈련기간 및 훈련시간, 훈련방법, 훈련 교사, 적용기간

□ 활용대상

- 「근로자 직업능력개발법」에 따른 직업능력개발 훈련
- 기타 직업교육훈련

□ 활용(예시)

- 국가직무능력표준에 따라 제시한 능력단위별 훈련기준을 조합하여 훈련기준으로 활용

<방법 1> 훈련이수체계도에서 제시한 훈련과정/과목으로 편성

<자동차차체정비 훈련 예시>

훈련수준	훈 련 모 들		구 분
	표준 직무	명 칭	
1수준(정비사)	자동차차체정비	단품교환	필수
		방음방침작업	

<방법 2> 훈련이수체계도에서 제시한 훈련과정/과목(필수)과 다른 직종의 훈련과정/과목(선택)으로 편성

자격종목	훈 련 모 들		구 분
	표준 직무	명 칭	
1수준(정비사)	자동차차체정비	단품교환	필수
		방음방침작업	
	자동차도장	건조작업	선택
		구도막제거작업	

1.1. 훈련기준

I. 개요

1. 직 종 명 : 응용SW엔지니어링

2. 직종 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 업무에 종사

3. 훈련이수체계(수준별 이수 과정/과목)

8	임원	SW아키텍처 미션 수립			보안계획 수립				
7	부장 차장	SW아키텍처 수행 관리 SW아키텍처 요구사항 명세화 후보 SW아키텍처 명세화 SW아키텍처 설계 SW아키텍처 문서화	애플리케이션 요구사항 분석	개념데이터 모델링	NW 종합 평가 NW 프로젝트 관리 NW 자원 관리 설계 NW QoS 제어 설계 NW 환경분석 NW 도출요지 설계 NW 프로토타입 분석 NW 프로토타입 설계	보안감사 수행	UI/UX 환경 분석 UI/UX 계획 수립	시스템SW 요구사항 분석 시스템SW 아키텍처 설계	
6	과장 대리	SW아키텍처 요소기술산정 (구) SW아키텍처설계 SW아키텍처 테스트 SW아키텍처 변경 관리	애플리케이션 디버깅 인터페이스 설계 기능모델 설계 애플리케이션 설계 정적모델 설계 동적모델 설계 소프트웨어공학 활용 소프트웨어 개발방법론 활용	디바이스 드라이버 분석 디바이스 드라이버 설계 운영체제 지식 디바이스 드라이버 구현	DB 요구사항분석 논리DB 설계 물리DB 설계 데이터 품질관리 DB 상용화 데이터 전환 설계	NW SW 아키텍처 수립	보안위협 평가 보안요구사항 정의 관리자 보안 구축 보안체계 운영 관리 보안인증 관리	UI/UX 컨셉 기획 UI 아키텍처 설계	시스템SW 상세 설계 시스템SW 테스트 시스템SW 기술준서 작성
5	사원	SW 개발 지원	요구사항확인 데이터 입출력 구현 통합 구현 정보시스템 이행 제품소프트웨어 패키징 서버프로그램 구현 인터페이스 구현 화면설계 애플리케이션 테스트관리	임베디드 시스템 테스트링 펌웨어 분석 펌웨어 설계 펌웨어 구현 운영체제 커널 분석 디바이스 드라이버 구현 환경 구축 임베디드 애플리케이션 분석 임베디드 애플리케이션 설계 임베디드 애플리케이션 구현	데이터표준화 SQL 응용	NW SW 개발 방법 수립	SW개발 보안 구축 DB 보안구축 시스템 보안 구축 NW 보안구축	사용자 리서치 UI/UX 요구 분석 UI/UX 가이드 제작	시스템SW 배포 시스템SW 통합 구현 시스템SW 단위 모듈 구현 시스템SW 인터페이스 구현
4	사원	SW아키텍처 이행	하드웨어 분석 기술문서 개발 펌웨어 구현 환경 구축 임베디드 애플리케이션 구현 환경 구축	DB구현			UI 디자인 UI 구현		
3	사원	프로그래밍 언어 활용 응용SW 기초기술 활용 애플리케이션 배포 화면 구현 애플리케이션 테스트 수행	오른 플랫폼 활용	SQL활용	NW 프로그래밍 구현	물리적 보안 구축 보안위협 관리통제	UI 테스트		
-		직업기초능력							
작업준비 능력	SW 아키텍처	응용SW 엔지니어링	임베디드SW 엔지니어링	DB 엔지니어링	NW엔지니어링	보안 엔지니어링	UI/UX 엔지니어링	시스템SW엔지 니어링	

4. 훈련시설

<div> <div>훈련인원</div> <div>시설명</div> </div>	기준인원	면 적	기준인원 초과 시 면적 적용	시 설 활용구분(공용/전용)
강의실	20명	45㎡	1명당 1.5㎡씩 추가	공 용
컴퓨터실	20명	45㎡	1명당 1.5㎡씩 추가	공 용
실 습 실	해당없음			-
공구·재료실	해당없음			-

※ 훈련시설은 훈련과정/과목에 필요한 시설을 구축

5. 교 사

○ 「근로자직업능력개발법」 제33조와 관련 규정에 따름

II. 훈련과정

○ 과정/과목명 : 직업기초능력

- 훈련개요

훈련목표	직업인으로서 갖추어야할 기본적인 소양을 함양
수 준	-
훈련시간	훈련과정의 전체교육시간 10%이내에서 자율편성
훈련가능시설	강의실 또는 컴퓨터실
권장훈련방법	집체 또는 원격훈련

- 편성내용

단 원 명	학 습 내 용
의사소통능력	업무를 수행함에 있어 글과 말을 읽고 들음으로써 다른 사람이 뜻한 바를 파악하고, 자기가 뜻한 바를 글과 말을 통해 정확하게 쓰거나 말하는 능력함양
수리능력	업무를 수행함에 있어 사칙연산, 통계, 확률의 의미를 정확하게 이해하고 이를 업무에 적용하는 능력함양
문제해결능력	업무를 수행함에 있어 문제 상황이 발생하였을 경우, 창조적이고 논리적인 사고를 통하여 이를 올바르게 인식하고 적절히 해결하는 능력함양
자기개발능력	업무를 추진하는데 스스로를 관리하고 개발하는 능력함양
자원관리능력	업무를 수행하는데 시간, 자본, 재료 및 시설, 인적자원 등의 자원 가운데 무엇이 얼마나 필요한지를 확인하고, 이용 가능한 자원을 최대한 수집하여 실제 업무에 어떻게 활용할 것인지를 계획하고, 계획대로 업무 수행에 이를 할당하는 능력
대인관계능력	업무를 수행하는데 있어 접촉하게 되는 사람들과 문제를 일으키지 않고 원만하게 지내는 능력
정보능력	업무와 관련된 정보를 수집하고, 이를 분석하여 의미 있는 정보를 찾아내며, 의미 있는 정보를 업무수행에 적절하도록 조직하고, 조직된 정보를 관리하며, 업무 수행에 이러한 정보를 활용하고, 이러한 제 과정에 컴퓨터를 사용하는 능력함양
기술능력	업무를 수행함에 있어 도구, 장치 등을 포함하여 필요한 기술에는 어떠한 것들이 있는지 이해하고, 실제로 업무를 수행함에 있어 적절한 기술을 선택하여, 적용하는 능력함양
조직이해능력	업무를 원활하게 수행하기 위해 국제적인 추세를 포함하여 조직의 체제와 경영에 대해 이해하는 능력함양
직업윤리	업무를 수행함에 있어 원만한 직업생활을 위해 필요한 태도, 매너, 올바른 직업관 함양

○ 과정/과목명 : 2001020201_16v3 요구사항 확인

- 훈련개요

훈련목표	업무 분석가가 수집·분석·정의한 요구사항과 이에 따른 분석모델에 대해서 확인과 현행 시스템에 대한 분석을 통해 개발하고자 하는 응용소프트웨어에 대한 이해를 높이는 능력을 함양
수 준	4
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
현행 시스템 분석하기	1.1 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다. 1.3 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 도메인에 대한 분석 • 요구사항 명세화 정확성 여부 • 서버 하드웨어 용량 산정 방법 숙지 및 수행 내역 • 운영체제의 주요 기능, 구조, 특성에 대한 분석 • DBMS의 주요 기능, 구조, 특성에 대한 분석 • WAS의 주요 기능, 구조, 특성에 대한 분석
요구사항 확인하기	2.1 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다. 2.2 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다. 2.3 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.	
분석모델 확인하기	3.1 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다. 3.2 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다. 3.3 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 가상화 관련 지식 • 소프트웨어 개발 방법론 • 업무 특성에 대한 이해 • 요구공학 방법론 • 요구분석기법 • 클라우드 컴퓨팅 관련 지식 • 타당성 분석기법 • 통계학 • 프로젝트 환경 및 특수성 • 프로젝트 환경 및 특수성 • 플랫폼에 따른 기능 및 성능 특성 • 해당 산업 분야에 대한 지식 • 해당 플랫폼에 대한 지식
기 술	<ul style="list-style-type: none"> • 가상화 관련 기술 • 내부 및 외부 환경 분석 기술 • 네트워크 구성 및 관리 능력 • 리뷰 진행 기술 • 분석 CASE 도구 사용 기술 • 저장장치 구성 및 관리 능력 • 요구사항 관리 도구 사용 기술 • 운영체제 구성 및 관리 기술 • 유즈케이스 작성 능력 • DBMS 구성 및 관리 기술 • UML 작성 기술
태 도	<ul style="list-style-type: none"> • 검증할 항목 분석을 위한 치밀한 태도 • 고객의 요청을 수용하고자하는 태도 • 기술 관련 각종 정보 수집에 대한 적극성 • 분류 및 정리 태도 • 비즈니스의 내용을 정확히 반영하는 프로세스 모델을 만들려는 치밀한 자세 • 시스템과 개발 소프트웨어와의 관계를 파악하려는 태도 • 시스템에 대한 정확성과 이해의 완전성을 갖고자 하는 태도 • 요구사항에 대한 일관성을 부여하는 자세 • 요구사항을 유기적 관계를 발견하는 자세 • 요구사항을 정확하게 기술하려는 자세 • 요구사항의 명확성을 찾으려는 자세 • 요구사항의 정확성과 완전성을 확보하려는 자세 • 용어정의 정확성과 완전성을 기하고자하는 의지 • 정확한 유스케이스를 이해하고 분석하려는 자세 • 책임감 및 검증에 대한 완벽함을 추구하는 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성 도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당 없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020206_16v4 통합 구현

- 훈련개요

훈련목표	모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하여 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력을 함양
수 준	4
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
연계 데이터 구성하기	1.1 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다. 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 연계 메커니즘에 대한 이해도 • 연계 표준/프로토콜에 대한 이해도 • 미들웨어, 인터페이스 소프트웨어에 대한 이해도 • 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도
연계 매커니즘 구성하기	2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다. 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다. 2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다. 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.	
내외부 연계 모듈 구현하기	3.1 구성된 연계 메커니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다. 3.2 구성된 연계 메커니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다. 3.3 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다. 3.4 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
-----	---------

지식	<ul style="list-style-type: none"> • 개발 방법론 이해 • 구현에 요구되는 아키텍처의 이해 • 네트워크 이해 • 데이터베이스 이해 • 분산 프로그래밍에 대한 이해 • 설계 모델링 기법 • 설계 산출물의 이해 • 소프트웨어 아키텍처에 대한 이해 • 소프트웨어 테스트에 대한 이해 • 업무 특성에 대한 이해 • 프로그래밍어 활용방법의 이해 • 해당 산업 분야에 대한 지식
기술	<ul style="list-style-type: none"> • IDE 도구 활용 능력 • IDE 및 개발환경 도구 활용 • UML 작성 기술 • 기술영역별 미들웨어/솔루션 활용 • 단위테스트 도구 활용기법 • 데이터 보안 기술 • 데이터 연계/이관 도구 활용 능력 • 설계 모델링 기술 • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 • 프로그램 디버깅 기법 • 형상관리 도구 활용 기법
태도	<ul style="list-style-type: none"> • 개발 수행에 요구되는 표준을 준수하려는 태도 • 개발 팀원 간의 원활한 협업을 추구하는 태도 • 고객의 요청을 적극적으로 수용하고자 하는 태도 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 산출물 완성도를 위한 적극적인 태도 • 성공적인 개발을 위한 의지 • 정확성과 완전성을 기하고자하는 의지 • 주어진 과제를 완수하는 책임감

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성 도구	세트	공용	1
• 빔 프로젝트	대	공용	30
• SW개발 도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020208_16v3 정보시스템 이행

- 훈련개요

훈련목표	개발자 환경에서 개발한 결과물을 운영 환경에 설치하고, 사용자 요구사항과 최종적으로 일치하는지에 대해 승인을 얻어 응용소프트웨어 결과물을 사용자에게 전달하여 인계하고 시스템을 운영할 수 있도록 교육하고 지원하는 능력을 함양
수 준	4
훈련시간	48시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
정보시스템 사용자 교육하기	1.1 개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다. 1.2 개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다. 1.3 효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검 할 수 있다. 1.4 교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육결과 및 성과를 평가할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 교육일정 계획이 실현가능하고, 교육시행 후 교육성과 • 작성된 매뉴얼(사용자, 개발자, 운영자 등)의 구체성과 이해 용이성 • 시스템 운영환경 설정과 애플리케이션의 배치 상황 • 이해 시간대별 영역별 사전점검표의 구체성 • 데이터 이행작업 점검 및 검증 • 이행여부 결정을 위한 과정 및 결정근거 • 이행 후 오류 및 조치 내역과 사용자 추가요구사항 수집 및 반영내역 • 시스템 안정화 및 성능개선을 위한 조치 내역
정보시스템 이행하기	2.1 하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다. 2.2 시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발할 수 있다. 2.3 정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증할 수 있다. 2.4 정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.	
정보시스템 안정화하기	3.1 정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다. 3.2 정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정보시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다. 3.3 개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 국제 표준 제품 품질특성에 대한 사항 • 문서작성 도구 활용 방법 • 소프트웨어 아키텍처에 대한 이해 • 시스템 성능 분석 및 진단에 대한 지식 • 시스템 운영 환경에 대한 이해 • 시스템 운영에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 • 업무 특성에 대한 이해 • 표준 양식(Template) 작성 및 활용법 • 피교육 집단 특성에 대한 이해 • 해당 산업 분야에 대한 지식
기 술	<ul style="list-style-type: none"> • 결함관리 도구 활용 능력 • 교육성과 분석 기법 • 교육성과 측정 기법 • 멀티미디어 작성 도구 활용 능력 • 문서작성 도구 활용 능력 • 성능 측정 도구 활용 능력 • 애플리케이션 모니터링 도구 활용 능력 • 애플리케이션 배포 도구 활용 능력 • 애플리케이션 패키징 기술 • 표준 용어집 • 프리젠테이션 기법 • 형상관리 도구 활용 기법
태 도	<ul style="list-style-type: none"> • 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 태도 • 교육효과를 높이기 위한 기자재 및 장소 확보 의지 • 교재의 일관성 확보를 위해 노력하는 태도 • 기술 및 제품 동향에 관한 각종 정보 수집에 대한 적극성 • 다양한 이해당사자의 의견을 긍정적으로 수용할 수 있는 태도 • 사용자 질의사항에 대한 긍정적 태도 • 성공적 시스템 이행을 위한 의지 • 시스템 성능의 문제점 및 성능 개선을 위한 의지 • 시스템 안정화를 위한 의지 • 오류 및 추가 요구사항에 대한 반영 의지 • 제품에 대한 책임감 및 불량에 대한 재발 방지를 위해 노력하는 태도 • 제품의 품질 및 고객 우선주의 태도 • 표준 양식 준수 의지

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성 도구	세트	공용	1
• 빔 프로젝트	대	공용	30
• SW개발 도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020209_16v4 제품소프트웨어 패키징

- 훈련개요

훈련목표	개발이 완료된 제품소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전관리를 수행하는 능력을 함양
수 준	4
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
제품소프트웨어 패키징하기	1.1 신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다. 1.2 이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다. 1.3 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 제품소프트웨어 기획 의도에 대한 이해 • 패키징 산출물 및 다이어그램 이해 • 패키징시 필요한 패키징, 복제방식, 저작권 룰 등에 대한 이해 • 형상관리 지침/도구에 대한 이해 • 버전관리 규칙의 준수 • 제품 SW UX/UX에 대한 이해 • 제품 SW UX/UX와 일관된 매뉴얼 기술 • 제품소프트웨어 실행 요구환경, 설치 환경에 대한 이해 • 소프트웨어 아키텍처와 일관성
제품소프트웨어 매뉴얼 작성하기	2.1 사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. 2.2 사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. 2.3 사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.	
제품소프트웨어 버전관리하기	3.1 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다. 3.2 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다. 3.3 버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 소프트웨어 버전관리 및 도구 사용에 대한 이해 • 장애 복구 처리 절차 • 각종 매뉴얼 작성법 • 국제 표준 제품 품질특성에 대한 사항 • 릴리즈 노트 작성법 • 사용자의 제품 실행 환경에 대한 이해 • 제품 패키징, 배포 관련 표준에 대한 지식 • 제품소프트웨어 관련 시장에 대한 지식 • 제품소프트웨어 사용에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 • 제품소프트웨어 아키텍처에 대한 이해 • 제품소프트웨어가 적용되는 업무 특성에 대한 이해
기 술	<ul style="list-style-type: none"> • API 문서 생성 기술(JavaDoc, doxygen) • Help File 생성 기술 • 소프트웨어 버전관리 도구 활용 능력 • 결함관리 도구 활용 능력 • 멀티미디어 작성 도구 활용 능력 • 문서작성 도구 활용 능력 • 빌드 자동화 도구 활용 능력 • 성능 측정 도구 활용 능력 • 암호화/보안 모듈 (DRM) 탑재 기술 • 애플리케이션 모니터링 도구 활용 능력 • 애플리케이션 배포 도구 활용 능력 • 애플리케이션 패키징 기술 • 표준 용어집 • 프리젠테이션 기법 • 협업도구 활용 능력 • 형상관리 도구 활용 기법
태 도	<ul style="list-style-type: none"> • 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 태도 • 교재의 일관성 확보를 위해 노력하는 태도 • 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도 • 사용자 중심으로 제품소프트웨어 활용에 대해 이해하고자 하는 태도 • 제품소프트웨어 버전관리의 중요성에 대한 이해도 • 제품소프트웨어에 변경의 필요성에 대해 이해하고자 하는 태도성 • 제품소프트웨어의 특성에 대해 이해하고자 하는 태도 • 제품의 품질 및 고객 우선주의 태도 • 표준 양식 준수 의지

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성 도구	세트	공용	1
• 빔 프로젝트	대	공용	30
• SW개발 도구	세트	공용	1
• SW패키징 도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020211_16v4 서버프로그램 구현

- 훈련개요

훈련목표	서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현하는 능력을 함양
수 준	4
훈련시간	48시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
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, 운영 정책에 부합하는 형상관리 환경을 구축할 수 있다.</p>	<p>- 평가자는 다음의 사항을 평가해야 한다.</p> <ul style="list-style-type: none"> • 분석/설계 산출물 및 다이어그램 분석 • 개발 애플리케이션의 주요 매커니즘 및 적용 개발언어 분석 능력 • 코드 이름규칙, 주석처리 방법 • 형상관리 절차 및 기준의 이해 • 서버프로그램 공통 리소스(예: 데이터베이스) 활용의 모듈화 이해
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> <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>	<ul style="list-style-type: none"> • 프로그램 언어별 데이터베이스 연동 방법 이해 • 프로그램 언어별 웹 서버프로그램 개발 방법의 이해 • 서버프로그램에서 적용하는 주요 모듈구성방법(레이어별 패턴)의 이해
서버 프로그램 구현하기	<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</p>	

	<p>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>	
배치 프로그램 구현하기	<p>4.1 애플리케이션 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 배치 프로그램 구현 기술에 부합하는 배치 프로그램을 구현 할 수 있다.</p> <p>4.2 목표 시스템을 구성하는 하위 시스템간의 연동 시, 안정적이고 안전하게 동작할 수 있는 배치 프로그램을 구현 할 수 있다.</p> <p>4.3 개발된 배치 프로그램을 테스트를 수행할 수 있다.</p>	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 결함 관리 • 구현에 요구되는 아키텍처 • 단위 업무 • 단위테스트 기술동향 및 도구 • 설계 산출물 해석방법 • 소프트웨어 테스트 • 프로그램 언어 활용방법 • 개발에 필요한 하드웨어 및 소프트웨어 사양 • 개발환경에 요구되는 아키텍처 • 구현에 요구되는 아키텍처 • 네트워크 개념 • 데이터베이스 개념 • 보안 관련 개념 • 분산 처리 개념 • 분산 프로그래밍 개념 • 서버 개발 프레임워크 개념 • 설계 산출물 해석방법 • 소프트웨어 테스트 • 프로그램 언어 활용방법

기 술	<ul style="list-style-type: none"> • IDE 도구 활용 능력 • 네트워크 활용 기술 • 단위테스트 도구 활용기법 • 보안도구 활용 능력 • 인터페이스 도구/기법 (API, EAI, WebService, 등) • 컴파일러 사용 능력 • 컴파일러 사용법 • 프레임워크 활용 기술 • 프로그래밍언어 활용 기술 • 프로그램 디버깅 기술 • 프로그램 코드 검토 기술 • 협업도구 활용 능력 • 형상관리 도구 활용 기법 • 형상관리 도구 활용 능력 • 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 • 단위테스트 도구 활용기법 • 모듈화 기법 • 인터페이스 도구/기법 (API, EAI, WebService, 등) • 재사용 기법 • 컴파일러 사용법 • 프로그램 디버깅 기법 • 프로그램 코드 검토 기법 • 형상관리 도구 활용 기법
태 도	<ul style="list-style-type: none"> • 개발 수행에 요구되는 표준을 준수하려는 태도 • 개발 팀원 간의 원활한 협업을 추구하는 태도 • 기술 및 도구에 대해 적극적으로 배우고자 하는 태도 • 산출물 완성도를 위한 적극적인 태도 • 성공적인 개발을 위한 의지 • 작업환경 개선 의지 • 적극적이고 세밀히 분석하는 자세 • 제품의 품질 및 고객 우선주의 태도 • 주어진 과제를 완수하는 책임감 • 개발 수행에 요구되는 표준을 준수하려는 태도 • 개발 팀원 간의 원활한 협업을 추구하는 태도 • 산출물 완성도를 위한 적극적인 태도 • 성공적인 개발을 위한 의지 • 적극적인 업무 분석 태도 • 주어진 과제를 완수하는 책임감

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020212_16v4 인터페이스 구현

- 훈련개요

훈련목표	인터페이스 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력을 함양
수 준	4
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
인터페이스 설계서 확인하기	<p>1.1 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다.</p> <p>1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다.</p> <p>1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.</p>	<p>- 평가자는 다음의 사항을 평가해야 한다.</p> <ul style="list-style-type: none"> • 연계 메커니즘에 대한 이해도 • 연계 표준/프로토콜에 대한 이해도 • 미들웨어, 인터페이스 소프트웨어에 대한 이해도 • 모니터링 도구 사용법 및 모니터링 산출물에 대한 이해도
인터페이스 기능 구현하기	<p>2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다.</p> <p>2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다.</p> <p>2.3 개발하고자 하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다.</p> <p>2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.</p>	
인터페이스 구현 검증하기	<p>3.1 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다.</p> <p>3.2 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다.</p>	

	3.3 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.	
--	--	--

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 개발 방법론 • 구현에 요구되는 아키텍처 • 네트워크 • 데이터베이스 • 디자인 패턴 • 분산 프로그래밍 • 산업 분야별 업무특성 • 설계 모델링 기법 • 설계 산출물 • 소프트웨어 아키텍처 • 소프트웨어 테스트
기 술	<ul style="list-style-type: none"> • IDE 도구 활용 능력 • IDE 및 개발환경 도구 활용 • UML 작성 기술 • 기술영역별 미들웨어/솔루션 활용 • 기술영역별 미들웨어/솔루션 활용 기술 • 단위테스트 도구 활용기법 • 데이터 보안 기술 • 데이터 연계/이관 도구 활용 능력 • 설계 모델링 기술 • 프레임워크 활용 • 프레임워크 활용 능력 • 프로그램 디버깅 기법 • 프로그램 언어 활용 능력 • 프로그램 코드검토 기법 • 형상관리 도구 활용 기법
태 도	<ul style="list-style-type: none"> • 개발 수행에 요구되는 표준을 준수하려는 태도 • 개발 팀원 간의 원활한 협업을 추구하는 태도 • 고객의 요청을 적극적으로 수용하고자 하는 태도 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 산출물 완성도를 위한 적극적인 태도 • 성공적인 개발을 위한 의지 • 정확성과 완전성을 기하고자 하는 의지 • 주어진 과제를 완수하는 책임감

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
● 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020214_16v4 애플리케이션 배포

- 훈련개요

훈련목표	애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포하는 능력을 함양
수 준	4
훈련시간	30 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
애플리케이션 배포 환경 구성하기	1.1 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다. 1.2 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다. 1.3 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다. 1.4 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 애플리케이션 개발 및 운영 환경에 대한 이해 여부 점검 • 애플리케이션 배포 환경 구성을 위한 관련 도구 설치 및 설정 가능 여부 평가 • 빌드 환경 구성에 따른 각종 도구 사용 능력 평가 • 소스코드 검증 기법에 대한 이해를 바탕으로 도구 사용 능력 점검
애플리케이션 소스 검증하기	2.1 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다. 2.2 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다. 2.3 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.	• 애플리케이션 빌드 및 배포 절차에 대한 이해 및 도구 사용 능력 확인 • 애플리케이션 빌드 및 배포 스크립트 작성 실습 및 점검 • 플리케이션 빌드 및 배포 절차에 따라 빌드 실행 능력 점검 • 애플리케이션 빌드 결과에 대한 분석 및 이해 능력 점검
애플리케이션 빌드하 기	3.1 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다. 3.2 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다. 3.3 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다.	• 애플리케이션 빌드 실패 시 원복 수행 절차 이해 및 실시 여부 확인

	<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>	
애플리케이션 배포하기	<p>4.1 애플리케이션 실행 환경에 대한 정보를 확인할 수 있다.</p> <p>4.2 애플리케이션 배포 절차에 따라 운영환경에 적용할 수 있다.</p> <p>4.3 애플리케이션 배포 후 정상적으로 작동하는지 여부를 확인할 수 있다.</p> <p>4.4 애플리케이션 배포 결과 문제가 발생했을 경우 적용 내용을 이전 상태로 복원할 수 있다.</p>	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 빌드 결과 분석 지식 • 소스코드 검증 기법 • 소스코드 검증 기준 • 소스코드 검증 도구 • 소스코드 동적 검증 분석 기법 • 소스코드 정적 검증 분석 기법 • 애플리케이션 빌드 스크립트 문법 • 애플리케이션 운영 플랫폼 정보 • 애플리케이션 운영 환경 배포 절차 • 애플리케이션 원복 절차 • 형상관리 지침 • 배포 도구 설치 및 사용법 • 애플리케이션 배포 환경 구성 지식 • 형상관리 도구 설치 및 사용법 • 애플리케이션 빌드(Build) 절차 • 형상관리 도구 사용법 • 애플리케이션 빌드 시스템 가이드

기 술	<ul style="list-style-type: none"> • 배포 도구 명령어 • 배포 스크립트 작성 기술 • 빌드 검증 기술 • 빌드 도구 명령어 • 빌드 스크립트 작성 기술 • 빌드 실행 기술 • 소스코드 검증 기술 • 소스코드 검증 도구 결과 분석 기술 • 소스코드 검증 도구 사용 기술 • 애플리케이션 빌드 도구 적용 기술 • 애플리케이션 운영 환경 적용 도구 활용 기술 • 애플리케이션 원복 실행 도구 사용법 • 애플리케이션 원복 절차 • 운영 배포 결과 확인 기법 • 환경 설정 기법 • 애플리케이션 운영 환경 배포 기술 • 소스코드 확보 기술 • 형상관리 도구 명령어
태 도	<ul style="list-style-type: none"> • 다양한 분석도구를 다루면서 효율적으로 활용하고자 하는 자세 • 문제 발생 애플리케이션에 대한 원상 복구 여부를 판단하는 자세 • 배포 시 문제 발생 시 즉시 원인 파악을 하고자 하는 자세 • 빌드 결과를 이해하고 문제 발생 시 즉시 조치하는 태도 • 빌드 실패 시 원인 분석을 실시하여 개발자에서 설명하는 자세 • 빌드 절차를 준수하여 성공적으로 빌드를 수행하고자하는 자세 • 빌드 진행 상태를 모니터링 하면서 결과를 확인하는 태도 • 소스코드 검증 결과 문제 발생시 해결하고자 하는 적극적인 자세 • 애플리케이션 배포 이전에 빌드 후 패키징 결과물의 이상 여부를 검증하는 자세 • 애플리케이션 배포 후 정상 작동 여부를 확인하는 자세 • 애플리케이션 원복 결정 시 즉시 절차에 따라 실행하는 태도 • 애플리케이션의 특성을 고려한 배포 환경을 구성하고자 하는 자세 • 자동화 도구 외 수작업으로 검토 및 리뷰를 통해 검증을 실시하는 자세 • 자동화 도구를 활용하여 소스코드를 검증하는 노력 • 자동화 도구를 효율적으로 다룰 수 있는 자세 • 애플리케이션 배포 이전에 정확한 환경 정보로 구성했는지 여부를 확인하는 태도 <p>검증이 완료된 소스코드는 빌드 시스템으로 정확하게 이관하는 태도</p> <ul style="list-style-type: none"> • 애플리케이션의 품질 향상을 위해 다양한 관점에서 소스코드를 검증하는 자세 • 다양한 도구와 연동을 통한 최적의 배포 환경을 구성하고자 하는 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록

- 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020215_15v3 프로그래밍 언어 활용

- 훈련개요

훈련목표	응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현하는 능력을 함양
수 준	3
훈련시간	80시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
기본문법 활용하기	1.1 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다. 1.2 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다. 1.3 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 프로그래밍 언어 활용에 대한 기본 소양 점검 • 프로그래밍 언어 특성 활용 능력 점검 • 프로그래밍 언어 라이브러리 적용을 위한 기본 능력 및 사고 점검 • 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검
언어특성 활용하기	2.1 프로그래밍 언어별 특성을 파악하고 설명할 수 있다. 2.2 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다. 2.3. 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용 할 수 있다.	
라이브러리 활용하기	3.1 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다. 3.2 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다. 3.3 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 객체지향 프로그래밍 언어 • 디버깅 기법 • 라이브러리 기능에 대한 이해 • 라이브러리 적용을 위한 기술(파일입출력, 데이터입출력, 예외 처리 등)에 대한 이해 • 선언형 언어 • 스크립트 언어 • 애플리케이션 요구사항에 대한 이해 • 절차적 프로그래밍 언어 • 코드 검토 기법 • 프로그래밍 언어 기능 구현을 위한 알고리즘 • 프로그래밍 언어 기본 문법 • 프로그래밍 언어 문법 오류
기 술	<ul style="list-style-type: none"> • IDE 도구 활용 능력 • 개발에 필요한 프로그래밍 언어 및 도구 활용 능력 • 라이브러리 검색 능력 • 라이브러리 설치 능력 • 라이브러리 적용 능력 • 라이브러리를 활용한 프로토타입 작성 능력 • 요구사항 구현에 필요한 라이브러리 선택 능력 • 컴파일러 사용 능력 • 코드 검토 능력 • 프로그램 디버깅 기술 • 프로그램 언어 활용 능력 • 프로그램 코드 검토 능력
태 도	<ul style="list-style-type: none"> • 기 작성된 코드를 분석하기 위한 치밀한 태도 • 기본 문법을 이해하고 활용하려는 적극적인 자세 • 라이브러리를 모듈화하려는 의지 • 라이브러리를 활용하는 적극적인 태도 • 라이브러리의 영향범위를 최소화하려는 태도 • 라이브러리의 특성을 이해하려는 의지 • 문법 오류를 이해하고 수정하려는 적극적인 태도 • 복잡한 구조 및 절차를 단순화하려는 의지 • 어플리케이션 구현의 방법과 절차를 문서화 하려는 의지 • 요구사항을 준수하려는 태도 • 코드 검토 결과를 수용하고 반영하려는 태도 • 프로그래밍 언어의 특성을 적용하려는 태도 • 프로그래밍 언어의 규칙과 관습을 준수하려는 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
● 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020216_15v3 응용 SW 기초 기술 활용

- 훈련개요

훈련목표	응용소프트웨어개발을 위하여 운영체제, 데이터베이스, 네트워크의 기초 기술을 적용하고 응용개발에 필요한 환경을 구축하는 능력을 함양
수 준	3
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
운영체제 기초 활용하기	1.1 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다. 1.2 CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다. 1.3 운영체제에서 제공하는 작업 우선순위 설정 방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 운영체제의 주요 기능, 구조, 특성에 대한 이해 • 데이터베이스의 기본 개념 및 관계형데이터베이스 테이블 구성에 대한 이해 • 네트워크의 기본 개념 및 OSI 7 Layer의 각 특성에 대한 이해 • 기반서버의 주요 기능, 구조, 특성에 대한 이해 • 오픈소스기반의 개발도구에 대한 이해
데이터베이스 기초 활용하기	2.1 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다. 2.2 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다. 2.3 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.	
네트워크 기초 활용하기	3.1 네트워크 계층구조에서 각 층의 역할을 설명할 수 있다. 3.2 응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다. 3.3 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다.	
기본 개발환경 구축하기	4.1 응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다. 4.2 응용개발에 필요한 개발도구를 설치하고 운용할 수 있다. 4.3 웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • CRUD 연산에 대한 이해 • DBMS 각 유형별 특징 • ER 다이어그램 작성 방법 • IP 주소 체계 • TCP/UDP 방식 개념 • Windows 운영체제 기본 명령어 • 각 운영체제 특징에 대한 이해 • 개발 프로그램 설치 방법 • 네트워크 7 계층의 개념 • 라이브러리 및 필요 패키지 설치 방법 • 리눅스 / 유닉스 계열 운영체제 기본 명령어 • 릴레이션의 특징 • 메모리 관리 기법의 이해 • 무결성 제약조건의 이해 • 버전관리 시스템 구축 • 운영체제 설치 및 제거 방법 • 운영체제 환경변수 설정 방법 • 인터넷 구성의 개념 • 테이블 선언 및 조작어 • 트랜스포트 방식 개념 • 프로세스 스케줄링 기법의 이해 • 프로토콜 개념 • 필요 응용 소프트웨어에 대한 이해 • 사용자의 프로그램 실행 환경에 대한 이해

기 술	<ul style="list-style-type: none"> • batch 파일, shell script 활용 능력 • E-R 다이어그램 작성 기술 • WINDOWS/UNIX/LINUX 인터넷 설정변경 기술 • 각 운영체제에 맞는 응용프로그램 선택 능력 • 개발 TOOL 사용 능력 • 개발 TOOL에 맞는 운영체제 선택 능력 • 개발에 필요한 프로그래밍 언어 선택 능력 • 개발환경에 인터넷 구축 기술 • 기본 명령어 활용 능력 • 데이터베이스 개발 TOOL 사용 능력 • 라우터 설정 기술 • 라이브러리 및 필요 패키지 선택 및 활용 능력 • 릴레이션 작성 기술 • 설계 모델링 기술 • 소프트웨어 버전관리 도구 활용 능력 • 오픈소스기반 DBMS 설치 기술 • 운영체제 오류 해결방법 검색 기술 • 운영체제 환경변수 설정 기술 • 인터넷 환경변수 설정 기술 • 작업우선순위 변경 기술 • 테이블 제작 및 관리 언어 활용 능력 • 필요 관련 정보 검색 기술 • 필요 프로그램 관련 정보 검색 기술
태 도	<ul style="list-style-type: none"> • 개발 수행에 요구되는 표준 절차를 준수하려는 태도 • 개발도구 관련 기술 및 제품에 대한 정보수집에 대한 적극성 • 개발도구 설치시 발생한 문제점을 해결하려는 의지 • 네트워크 관련 기술 및 제품에 대한 정보수집에 대한 적극성 • 네트워크 설정 관련 문제점을 해결하려는 의지 • 능동적 태도 • 다양한 운영체제의 특징을 이해하려는 태도 • 데이터베이스 관련 기술 및 제품에 대한 정보수집에 대한 적극성 • 문제점을 해결하려는 의지 • 사용할 프로그램의 특성을 이해하려는 태도 • 상황에 맞게 데이터베이스를 작성하고 변경하는 능동적인 태도 • 오픈소스운동에 참여하려는 의지 • 운영체제 관련 기술 및 제품에 대한 정보수집에 대한 적극성 • 운영체제 활용 중 발생한 문제를 해결하고자 하는 의지 • 자신이 수행한 작업에 대한 평가의 객관성 • 적극적이고 세밀히 각 단계를 분석하는 자세 • 정규화 과정을 지속적으로 관찰하는 태도 • 주어진 문제를 완수하는 책임감 • 콘솔/터미널을 활용하려는 태도 • 활용에 필요한 기술을 이해하려는 의지

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1

• SW개발도구	세트	공용	1
----------	----	----	---

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020217_16v4 애플리케이션 리팩토링

- 훈련개요

훈련목표	애플리케이션 리팩토링이란 소스코드가 수행하는 기능을 유지하면서 코드 가독성 및 잠재적 결함을 제거하도록 코드의 구조를 개선하는 능력을 함양
수 준	5
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
리팩토링 기준 수립하기	1.1 리팩토링을 적용하여 달성하려는 품질 목표를 수립할 수 있다. 1.2 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다. 1.3 리팩토링에 활용할 수 있는 도구 및 절차를 수립할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 리팩토링 전/후의 기능성 유지 여부 확인 • 분석 대상에서 발견한 개선 점의 정확성 여부 확인 • 적용한 패턴의 정확성 확인
소스 코드 분석하기	2.1 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다. 2.2 성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별할 수 있다. 2.3 잠재 결함 요소를 내포한 소스 코드를 식별할 수 있다.	
리팩토링 적용하기	3.1 리팩토링을 위한 소스 코드 구조를 개선할 수 있다. 3.2 개선한 소스 코드가 초기 기능을 유지하는지 확인할 수 있다. 3.3 개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 국제 코딩 표준 • 디자인 패턴 • 리팩토링 기법 • 리팩토링 적용 기법 • 비기능적 품질 특성 • 비기능적 품질 특성 충족 기법 • 코드 스멜 유형 • 클린 코드 특징 • 클린 코드의 필요성 및 특징 • 회귀 테스트

기 술	<ul style="list-style-type: none"> • IDE 도구 활용 능력 • 리팩토링 도구 활용 능력 • 설계 패턴 적용 기술 • 역공학 도구 활용 기술 • 좋은 설계 (Good Design) 기준 • 체크리스트 작성 기술 • 코드 동적 분석 도구 활용 기술 • 코드 인스펙션 기술 • 코드 정적 분석 도구 활용 기술 • 테스트 도구 활용 기술 • 프로그램 디버깅 능력
태 도	<ul style="list-style-type: none"> • 결함 없는 코드로 개선하려는 태도 • 이해하기 쉽고 확장하기 쉬운 코드로 개선하려는 태도 • 지속적으로 리팩토링을 수행하려는 적극적 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020218_16v4 인터페이스 설계

- 훈련개요

훈련목표	인터페이스 설계란 응용소프트웨어 개발을 위해 정의된 시스템 인터페이스 요구사항을 확인하고 인터페이스 대상을 식별하여 인터페이스 설계서를 작성하는 능력을 함양
수 준	5
훈련시간	40시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
인터페이스 요구사항 확인하기	1.1 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 비 기능 요구사항을 분석할 수 있다. 1.2 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 기능 요구사항을 분석할 수 있다. 1.3 개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의 정확성과 완전성을 확인한다.	- 평가자는 다음의 사항을 평가해야 한다. • 인터페이스 요구사항 분석 및 확인 절차 • 인터페이스 대상 시스템 및 연계 시스템 식별 과정 • 인터페이스 대상 송수신 데이터 식별 및 구성 • 내외부 시스템간 인터페이스 설계서 작성 절차 • 소프트웨어 아키텍처와 일관성 여부
인터페이스 대상 식별하기	2.1 개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대상 시스템을 식별할 수 있다. 2.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 연계 시스템을 식별할 수 있다. 2.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 송수신 데이터를 식별할 수 있다.	
인터페이스 상세 설계하기	3.1 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위한 송수신 방법을 명세화할 수 있다. 3.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 데이터를 명세화할 수 있다. 3.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오류 시 처리방안을 명세화할 수 있다. 3.4 소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
-----	---------

지식	<ul style="list-style-type: none"> • 네트워크 • 산업 분야별 업무특성 • 설계 모델링 방법론 • 소프트웨어 아키텍처 • 시스템 아키텍처 • 요구공학 • 인터페이스 시스템
기술	<ul style="list-style-type: none"> • IDE 및 개발환경 도구 활용 • UML 작성 기술 • 기술영역별 미들웨어와 솔루션 활용 능력 • 리뷰 진행 기술 • 설계 모델링 기술 • 요구사항 관리 도구 사용 능력 • 프레임워크 활용
태도	<ul style="list-style-type: none"> • 사용자의 요구사항에 대한 적극적 수용 태도 • 산출물 완성도 확보를 위한 적극적인 자세 • 이해 당사자와의 협업을 위한 적극적 의사소통 • 정확한 인터페이스 시스템 식별을 위한 의지 • 조직의 개발방법론을 준수하려는 태도 • 조직의 품질관리 기준을 준수하려는 태도 • 정확한 요구사항의 이해를 위한 분석 자세

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020219_16v4 애플리케이션 요구사항 분석

- 훈련개요

훈련목표	애플리케이션 요구사항 분석이란 구현하고자 하는 애플리케이션의 요구사항을 도출, 분석, 명세화 및 요구사항 검증을 수행하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
요구사항 도출하기	<p>1.1 비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구사항 도출을 위한 정보를 수집 할 수 있다.</p> <p>1.2 인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다.</p> <p>1.3 요구사항 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.</p>	<p>- 평가자는 다음의 사항을 평가해야 한다.</p> <ul style="list-style-type: none"> 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. 요구사항 도출 기법에 대한 이해 사용자 요구사항 정의서 작성 방법 사용되는 시스템이나 하드웨어에 대한 이해 사용되는 소프트웨어에 대한 이해
요구사항 분석하기	<p>2.1 요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구사항과 비기능적 요구사항을 분류 할 수 있다.</p> <p>2.2 도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다.</p> <p>2.3 도출한 요구사항이 개발하고자 하는 애플리케이션 전체 또는 단위 컴포넌트에 향에 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다.</p> <p>2.4 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.</p>	<ul style="list-style-type: none"> 소프트웨어 요구사항 명세서 작성 방법에 대한 이해 도출한 요구사항에 대한 기준 수립에 대한 이해 요구사항 명세서에 대한 검증 및 검토에 대한 이해
요구사항 명세화하기	<p>3.1 애플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의서를 작성할 수 있다.</p>	

	<p>3.2 구현하고자 하는 애플리케이션이 구동되는 시스템이나 하드웨어 관련 내용을 구체화 하여 시스템 요구사항 문서를 작성할 수 있다.</p> <p>3.3 구현하고자 하는 애플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)을 작성할 수 있다.</p>	
요구사항 검증하기	<p>4.1 소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다.</p> <p>4.2 작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한 설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다.</p> <p>4.3 기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다.</p> <p>4.4 분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.</p>	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • SW 제품 개발 수명 주기 이론 • 요구 공학(Requirement Engineering) 이론 • SW 제품 요구사항 관리 절차 • SW 제품 요구사항 도출 지식 • SW 제품 요구사항 정의 기준 • 요구사항 명세서(Software Requirement Specification) 작성 기준 • IEEE830 (IEEE Recommended Practice for Software Requirements pecifications) • 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 • BABOK (Business Analysis Body of Knowledge) • SWEBOK(Software Engineering Body of Knowledge) • SW제품 개발 요구 공학 • SW제품 품질 속성 항목 • SW제품 품질 평가 항목 • 정보시스템의 기술적 지식 • 요구사항 적용 범위 확인 지식 • 요구사항 우선 순위 결정 기준 • 요구사항 영향범위 설정 기준

기 술	<ul style="list-style-type: none"> • 요구사항 도출 기법 • 인터뷰 기법 • 시나리오 작성 기법 • 프로토타입 제작 기법 • 그룹 회의 토의 진행 기법 • 사용자 스토리 작성 기법 • 요구사항 기록 방법 • 요구사항 분석 능력 • 요구사항 분류 기술 • 요구사항의 우선순위를 식별하는 능력 • 사용자와 협상 능력 • 상위레벨 SW제품 아키텍처 수립 기술
태 도	<ul style="list-style-type: none"> • 도출한 요구사항을 사용자 관점에서 정확하게 이해하고자 하는 태도 • 다양한 출처로 부터의 요구사항을 명확하게 도출하고자 노력하는 태도 • 요구사항 도출 후 정확하게 기록하고자 하는 태도 • 구체적인 요구사항 도출을 위해 다양한 기법을 활용하고자 하는 열린 자세 • 고객관점의 요구사항을 이해하려는 태도 • 요구 분석 내용을 체계적으로 이해하고자 하는 태도 • 요구사항을 명확하게 하기 위해 사용자와 협상하는 태도 • 요구사항 분석 기준을 통해 우선순위를 결정하는 태도 • 요구사항 영향범위를 명확히하고자 하는 태도 • 요구사항 정의서 양식에 맞추어 체계적으로 작성하고자 하는 태도 • 개발 제품의 요건을 명확하게 도출 할 수 있는 태도 • 구체적이고 명확한 요구사항 정의서를 작성하고자 하는 태도 • 요구사항 명세서의 내용을 고객과 개발팀이 이해할 수 있도록 기술하는 노력 • 요구사항 명세서의 일관성과 완성도를 검증하고자 하는 자세 • 요구사항 명세서의 오류를 발견하여 조치하고자 하는 태도 • 요구사항의 구현 가능성, 안정성, 기술적 환경 분석 자세 • 사용자 요구사항 반영 여부를 확인하고자 하는 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020220_16v1 기능모델 설계

- 훈련개요

훈련목표	기능모델 설계란 애플리케이션 요구사항에 의해 도출, 분석된 내용을 애플리케이션의 기능으로 명세화하여 요구사항을 검증하고 소프트웨어 개발범위를 확정하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
기능모델 작성하기	1.1 기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다. 1.2 기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별하여 다이어그램을 작성할 수 있다. 1.3 기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능모델 설계 명세서를 작성할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • 요구사항 도출 기법에 대한 이해
기능모델 검증하기	2.1 도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다. 2.2 작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다. 2.3 작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함하고 있는지 검증할 수 있다.	• 사용자 요구사항 정의서 작성 방법 • 사용되는 시스템이나 하드웨어에 대한 이해 • 사용되는 소프트웨어에 대한 이해 • 소프트웨어 요구사항 명세서 작성 방법에 대한 이해 • 도출한 요구사항에 대한 기준 수립에 대한 이해 • 요구사항 명세서에 대한 검증 및 검토에 대한 이해

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 요구 공학(Requirement Engineering) 이론 • SWEBOK(Software Engineering Body of Knowledge)

기 술	<ul style="list-style-type: none"> • UML 작성 기술 • 모델링 기술 • 모델링 명세화 기술 • 모델 검증 기술 • 의사소통 기술
태 도	<ul style="list-style-type: none"> • 고객의 요청을 적극적으로 수용하고자 하는 의지 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 기능모델의 일관성과 완성도를 검증하고자 하는 자세 • 기능의 누락이나 오류를 발견하여 조치하고자 하는 태도 • 산출물 완성도 확보를 위한 적극적인 태도 • 고객과의 지속적인 소통을 위한 책임감 있는 자세

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020221_16v4 애플리케이션 설계

- 훈련개요

훈련목표	애플리케이션 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통모듈 설계, 타 시스템 연동에 대하여 상세 설계하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
공통 모듈 설계하기	1.1 재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다. 1.2 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다. 1.3 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다. 1.4 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • 요구사항 추적성 확보 • 소프트웨어 아키텍처와 일관성 확인 • 프로토타입 작성 및 검증 능력
타 시스템 연동설계하기	2.1 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다. 2.2 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다. 2.3 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어/솔루션에 대하여 명세를 작성할 수 있다. 2.4 소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
-----	---------

지식	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 소프트웨어 아키텍처에 대한 이해 • 설계 모델링 기법 • 네트워크 이해 • 분산 프로그래밍에 대한 이해
기술	<ul style="list-style-type: none"> • UML 작성 기술 • 설계 모델링 기술 • E-R 모델 작성 기술 • IDE 및 개발환경 도구 활용 • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 • 기술영역별 미들웨어/솔루션 활용
태도	<ul style="list-style-type: none"> • 고객의 요청을 적극적으로 수용하고자 하는 의지 • 책임감 및 설계문서에 완벽함을 추구하는 태도 • 아키텍트와 설계자간의 협업을 추구하는 태도 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 산출물 완성도를 위한 적극적인 태도 • 주어진 과제를 완수하는 책임감

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020222_16v4 정적모델 설계

- 훈련개요

훈련목표	정적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 정적모델을 설계하고 검증하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
정적 분석모델 검증하기	1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내역을 확인할 수 있다. 1.2 분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • 요구사항 추적성 확보 • 소프트웨어 아키텍처와 일관성 확인 • 프로토타입 작성 및 검증 능력
정적모델 상세화하기	2.1 분석된 정적모델을 정적 설계모델로 변환하기 위해 변환 규칙을 정의할 수 있다. 2.2 정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다. 2.3 상세화된 정적 설계모델의 명세서를 작성할 수 있다. 2.4 설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 소프트웨어 아키텍처에 대한 이해 • 설계 모델링 기법 • 네트워크 이해 • 데이터베이스 이해
기 술	<ul style="list-style-type: none"> • UML 작성 기술 • 설계 모델링 기술 • E-R 모델 작성 기술 • IDE 및 개발환경 도구 활용 • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 • 기술영역별 미들웨어/솔루션 활용

태도	<ul style="list-style-type: none"> • 고객의 요청을 적극적으로 수용하고자 하는 의지 • 책임감 및 설계문서에 완벽함을 추구하는 태도 • 아키텍트와 설계자간의 협업을 추구하는 태도 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 산출물 완성도를 위한 적극적인 태도 • 주어진 과제를 완수하는 책임감
----	---

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020223_16v4 동적모델 설계

- 훈련개요

훈련목표	동적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 동적모델을 설계하고 검증하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
동적 분석모델 검증하기	1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다. 1.2 분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	<ul style="list-style-type: none"> - 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • 요구사항 추적성 확보 • 소프트웨어 아키텍처와 일관성 확인 • 프로토타입 작성 및 검증 능력
동적모델 상세화하기	2.1 검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다. 2.2 동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다. 2.3 상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다. 2.4 도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.	

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 개발 방법론 이해 • 소프트웨어 아키텍처에 대한 이해 • 설계 모델링 기법 • 네트워크 이해 • 분산 프로그래밍에 대한 이해
기 술	<ul style="list-style-type: none"> • UML 작성 기술 • 설계 모델링 기술 • E-R 모델 작성 기술 • IDE 및 개발환경 도구 활용 • 프레임워크(애플리케이션, 웹, 데이터 액세스, 테스트 자동화, 기타) 활용 • 기술영역별 미들웨어/솔루션 활용

태도	<ul style="list-style-type: none"> • 고객의 요청을 적극적으로 수용하고자 하는 의지 • 책임감 및 설계문서에 완벽함을 추구하는 태도 • 아키텍트와 설계자간의 협업을 추구하는 태도 • 도출된 모듈/기능 컴포넌트를 분석하기 위한 치밀한 태도 • 산출물 완성도를 위한 적극적인 태도 • 주어진 과제를 완수하는 책임감
----	---

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020224_16v4 화면 설계

- 훈련개요

훈련목표	화면 설계란 요구사항분석 단계에서 파악된 화면에 대한 요구사항을 소프트웨어 아키텍처 단계에서 정의된 구현 지침 및 UI/UX 엔지니어가 제시한 UI표준과 지침에 따라 화면을 설계하는 능력을 함양
수 준	5
훈련시간	48 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
UI 요구사항 확인하기	1.1 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다. 1.2 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다. 1.3 작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • UI 설계서에 대한 이해도 • UI 표준 및 지침에 대한 이해도
UI 설계하기	2.1 UI 요구사항과 UI 표준 및 지침에 따라, 화면과 폼의 흐름을 설계하고, 제약사항을 화면과 폼 흐름 설계에 반영할 수 있다. 2.2 UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다. 2.3 UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 폼을 설계할 수 있다.	• 최신 UI 기술들에 대한 이해도 • 사용자 경험 분석에 대한 이해도 • 프로토타입 작성 및 검증 능력 • 소프트웨어 아키텍처 상세구현 지침에 대한 이해도 • UI 설계를 소프트웨어 아키텍처 상세구현 지침과 UI 표준 및 지침에 따라 개발하는 능력

- 지식·기술·태도

구 분	주 요 내 용
-----	---------

지식	<ul style="list-style-type: none"> • 구현에 요구되는 아키텍처의 이해 • 유용성 개념 및 적용원리 • UI 기획/설계/구현 절차 • UI 표준 및 지침 • 유즈케이스에 대한 이해 • 데이터 타입 특성 • 사용자 기반 메뉴 구조 작성 방법 • 감성공학에 대한 이해 • 프로토타입 작성법
기술	<ul style="list-style-type: none"> • 유즈케이스 작성 능력 • 스토리보드 작성 기술 • 프로토타입 도구 사용 능력 • IDE 도구 활용 능력 • UI 작성 그래픽 도구 활용 능력 • UI 설계도구 활용 능력 • UI 네비게이션 작성 도구 활용 능력 • 화면/폼 구성 도구 활용 능력
태도	<ul style="list-style-type: none"> • 고객을 적극적으로 이해하고 수용하기 위한 자세 • UI 표준 및 지침을 준수하려는 태도 • UI 관련자들과 원활한 관계를 유지하려는 태도 • 기술적 위험에 적극적으로 대비하려는 노력 • UI 설계 변경 요구 피드백에 대해 열린 마음으로 듣는 태도 • 고품질의 소프트웨어 개발에 대한 의지

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020225_16v4 화면 구현

- 훈련개요

훈련목표	화면 구현이란 UI 요구사항을 확인하여 설계한 UI 설계를 기반으로 화면을 구현하는 능력을 함양
수 준	3
훈련시간	48 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
UI 설계 확인하기	1.1 설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다. 1.2 UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다. 1.3 구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • UI 요건에 대한 이해도
UI 구현하기	2.1 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다. 2.2 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다. 2.3 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다. 2.4 구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	• UI 설계서에 대한 이해도 • UI 표준 및 지침에 대한 이해도 • 최신 UI 기술들에 대한 이해도 • 사용자 경험 분석에 대한 이해도 • 프로토타입 작성 및 검증 능력 • 소프트웨어 아키텍처 상세 구현 지침에 대한 이해도 • UI 설계를 소프트웨어 아키텍처 상세 구현 지침과 UI 표준 및 지침에 따라 개발하는 능력

- 지식·기술·태도

구 분	주 요 내 용
-----	---------

지식	<ul style="list-style-type: none"> 구현에 요구되는 아키텍처의 이해 유용성 개념 및 적용원리 UI 기획/설계/구현 절차 UI 표준 및 지침 유즈케이스에 대한 이해 데이터 타입 특성 사용자 기반 메뉴 구조 작성 방법 감성공학에 대한 이해 HTML 개념과 적용원리 CSS의 개념과 적용원리 자바스크립트의 개념과 적용원리 설계 산출물의 이해
기술	<ul style="list-style-type: none"> 유즈케이스 작성 능력 IDE 도구 활용 능력 UI 작성 그래픽 도구 활용 능력 HTML, CSS, 자바스크립트 등 표준기술 활용 능력 UI 네비게이션 작성 도구 활용 능력 화면/폼 구성 도구 활용 능력 화면 구성 능력 UI 작성 그래픽 도구 활용 능력 UI Framework(jQuery, AngularJS등) 활용 능력
태도	<ul style="list-style-type: none"> 고객을 적극적으로 이해하고 수용하기 위한 자세 UI 표준 및 지침을 준수하려는 태도 UI 관련자들과 원활한 관계를 유지하려는 태도 기술적 위험에 적극적으로 대비하려는 노력 UI 설계 변경 요구 피드백에 대해 열린 마음으로 듣는 태도 고품질의 소프트웨어 개발에 대한 의지

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020226_16v4 애플리케이션 테스트 관리

- 훈련개요

훈련목표	애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력을 함양
수 준	5
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
애플리케이션 테스트 케이스 설계하기	1.1 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다. 1.2 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다. 1.3 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • UI 요건에 대한 이해도 • UI 설계서에 대한 이해도 • UI 표준 및 지침에 대한 이해도 • 최신 UI 기술들에 대한 이해도
애플리케이션 개발자 통합 테스트하기	2.1 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. 2.2 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다. 2.3 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.	• 사용자 경험 분석에 대한 이해도 • 프로토타입 작성 및 검증 능력 • 소프트웨어 아키텍처 상세구현 지침에 대한 이해도 • UI 설계를 소프트웨어 아키텍처 상세구현 지침과 UI 표준 및 지침에 따라 개발하는 능력

- 지식 · 기술 · 태도

구 분	주 요 내 용
-----	---------

지식	<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 • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 소프트웨어 아키텍처에 대한 이해 • 국제 표준 제품 품질특성에 대한 사항 • 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)에 대한 지식 • 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법 • 결함 관리에 대한 지식 • 알고리즘 • 자료구조
기술	<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 • 소스코드 인스펙션 능력 • 소스코드 품질분석 도구(PMD, SONAR 등)
태도	<ul style="list-style-type: none"> • 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 자세 • 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

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020227_16v4 애플리케이션 테스트 수행

- 훈련개요

훈련목표	애플리케이션 테스트 수행이란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 분석된 테스트 케이스에 따라 테스트를 수행하고 결함을 조치하는능력을 함양
수 준	3
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
애플리케이션 테스트 수행하기	<p>1.1 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족 하는지에 대한 테스트를 수행할 수 있다.</p> <p>1.2 애플리케이션 테스트 수행으로 발견된 결함 을 유형별로 기록 할 수 있다.</p> <p>1.3 애플리케이션 테스트 수행 결과 발견된 결함 에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.</p>	<p>- 평가자는 다음의 사항을 평가해야 한다.</p> <ul style="list-style-type: none"> • 테스트 계획의 구성항목을 이해 하고 테스트 케이스 작성 능력 • 테스트 전략을 이해하고 테스트 케이스 작성 시 적용할 수 있는 능력 • 테스트 케이스를 이해하고 테스트를 수행할 수 있는 능력
애플리케이션 결함 조치하기	<p>2.1 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별 하고 조치에 대한 우선순위를 결정하고 적용할 수 있다.</p> <p>2.2 결함이 발생한 소스를 분석하고 기존에 구현 된 로직과의 연관성을 고려하여 부작용이 최소화 되도록 결함을 제거할 수 있다.</p> <p>2.3 애플리케이션 테스트 결과 결함 조치로 변경 되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.</p>	<ul style="list-style-type: none"> • 테스트 관련 국제 표준 지식체계에 대하여 이해하고 숙지하고 있는지의 여부

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<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 • 해당 산업 분야에 대한 지식 • 업무 특성에 대한 이해 • 소프트웨어 아키텍처에 대한 이해 • 국제 표준 제품 품질특성에 대한 사항 • 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)에 대한 지식 • 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법 • 결함 관리에 대한 지식
기 술	<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
태 도	<ul style="list-style-type: none"> • 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 자세 • 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

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020228_16v4 소프트웨어공학 활용

- 훈련개요

훈련목표	소프트웨어공학 활용이란 응용 소프트웨어 개발과 프로세스 적용활동의 관련된 지식을 소프트웨어의 완전성을 보장하고, 소프트웨어 품질을 평가하기 위해 CASE 도구와 형상관리를 통해 소프트웨어 공학 기술을 적용하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
CASE 도구 활용하기	1.1 개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다. 1.2 CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다. 1.3 CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다. • 소프트웨어 공학기술에 대한
애플리케이션 결함 조치하기	2.1 요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다. 2.2 요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다. 2.3 개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.	기본 소양 점검 • 소프트웨어 개발방법론을 적용하기 위한 기본 능력 및 사고 점검 • 관련 업무 영역의 개선 수행을 위한 소양 점검 • 관련 업무 경험의 실효성/충분성 점검 • 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 소프트웨어 개발 방법론 • 프로젝트 관리 방법론 • 소프트웨어 개발 표준 • 요구공학 방법론 • 비용산정 모델 • 아키텍처 평가 • 소프트웨어 개발 프레임워크에 대한 지식 • 국제 표준 제품 품질특성에 대한 사항 • 품질 점검 기법(리뷰, 워크스루, 인스펙션 등) • 품질 관리 프로세스 • 소프트웨어 개발 표준 • 시정 조치 관리 프로세스 • 감사 프로세스
기 술	<ul style="list-style-type: none"> • 협상 기술 • 의사결정 기술 • 프로세스 테일러링 기술 • UML 작성기술 • 모델링 기법 • 유즈케이스 모델 정의할 수 있는 기술 • 형상관리 기법 • 리뷰 진행 기술 • 인스펙션 진행 기술 • 감사 기술 • 요구사항명세서 작성기술 • 소프트웨어 요구사항 품질 측정, 점검 기술
태 도	<ul style="list-style-type: none"> • 새로운 공학도구를 사용하고 하는 도전의식 • 소프트웨어 개발방법론을 적용하기 위해 프로젝트를 분석하려는 자세 • 표준을 이해하고 준수하려는 의지 • 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도 • 소프트웨어의 품질에 대해 이해하고자 하는 태도 • 고객 및 제품 요구사항에 대하여 적극적으로 분석하는 자세 • 객관적인 평가방법을 작성하려는 객관성 • 품질요구사항에 대해 철저히 분석하려는 태도 • 품질요구사항에 대해 정확한 기준을 제시하려는 태도 • 품질 측정의 근거를 확보하기 위해 노력하는 태도 • 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도

- 장비

장 비 명	단 위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재 료 목 록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

○ 과정/과목명 : 2001020229_16v4 소프트웨어개발 방법론 활용

- 훈련개요

훈련목표	소프트웨어개발방법론 활용이란 응용소프트웨어 특성에 따라 정형화된 개발방법론을 선정하고, 프로젝트 특성에 맞도록 테일러링 하여 응용소프트웨어 개발에 활용하는 능력을 함양
수 준	6
훈련시간	40 시간
훈련가능시설	컴퓨터실
권장훈련방법	집체교육 또는 원격교육

- 편성내용

단 원 명 (능력단위 요소명)	훈 련 내 용 (수행준거)	평가시 고려사항
소프트웨어 개발방법론 선정하기	1.1 개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된 개발방법론 선정기준으로 활용할 수 있다. 1.2 정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.	- 평가자는 다음의 사항을 평가해야 한다. • 수행준거에 제시되어 있는 내용을 성공적으로 수행할 수 있는지를 평가해야 한다.
소프트웨어 개발방법론 테일러링하기	2.1 프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다. 2.2 확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다. 2.3 응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다. 2.4 테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다.	• 소프트웨어 공학기술에 대한 기본 소양 점검 • 소프트웨어 개발방법론을 적용하기 위한 기본 능력 및 사고 점검 • 관련 업무 영역의 개선 수행을 위한 소양 점검 • 관련 업무 경험의 실효성/충분성 점검 • 능력단위요소에 적절한 도구를 활용할 수 있는지를 점검

- 지식 · 기술 · 태도

구 분	주 요 내 용
지 식	<ul style="list-style-type: none"> • 소프트웨어 생명주기 모델 • 소프트웨어 개발 방법론 • 요구공학 방법론 • 비용산정 모델 • 반복, 점증 프로세스 • 소프트웨어 개발 프레임워크에 대한 지식
기 술	<ul style="list-style-type: none"> • 의사결정 기술 • 모델링 기법 • 모델링 도구 활용기술 • 협상기술 • 프로세스 테일러링 기술

태도	<ul style="list-style-type: none"> • 새로운 공학도구를 사용하고 하는 도전의식 • 소프트웨어 개발방법론을 적용하기 위해 프로젝트를 분석하려는 자세 • 소프트웨어의 품질에 대해 이해하고자 하는 태도 • 문장 구성력 및 시각적 표현력을 갖추고자 노력하는 태도 • 소프트웨어의 품질에 대해 이해하고자 하는 태도
----	---

- 장비

장비명	단위	활용구분(공용/전용)	1대당 활용인원
• 컴퓨터	대	공용	1
• 문서작성도구	세트	공용	1
• SW개발도구	세트	공용	1

※ 장비는 주장비만 제시한 것으로 그 외의 장비와 공구는 별도로 확보

- 재료

재료 목록
• 해당없음

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

재료 목록

※ 재료는 주재료만 제시한 것으로 그 외의 재료는 별도로 확보

Ⅲ. 고려사항

1. 활용방법

- 훈련기준에서 제시한 이외의 과정수립에 필요한 사항은 「근로자직업능력개발법」 등 관련 규정을 참고하시기 바랍니다.
- 본 훈련기준의 훈련과정은 모듈식으로, 장-단기과정 모두에서 활용가능하며, 훈련사업별로 요구하는 훈련과정 편성지침에 따라 편성할 수 있습니다.
- 3월 350시간 이상의 장기 훈련과정을 편성하는 경우, 수강생의 수준에 적합하게 훈련이수체계도에서 제시한 해당직종의 훈련과정/과목을 필수로 반영하고, 이외 관련 직종의 과정/과목을 선택하여 편성할 수 있습니다.
 - * 단, 훈련생이 ‘필수과정’의 일부 훈련 과정/과목을 이수하거나, 직무수행경력이 있는 경우에는 해당 훈련과정/과목을 제외하고 훈련할 수 있습니다.
 - * 효율적으로 훈련하기 위해 둘 이상의 과정/과목을 결합하여 대(大)과목으로 편성하거나, 하나의 과정/과목을 둘 이상의 세(細)과목으로 편성하여 훈련할 수 있습니다.
 - * 훈련과정/과목에서 제시한 훈련시간은 훈련생의 학습능력을 고려하여 최대 50%까지 연장하여 훈련할 수 있습니다.

2. 참고사항

가. 관련자격종목

- 정보관리기술사 ○ 컴퓨터시스템응용기술사 ○ 정보처리(산업)기사
- 정보처리기능사 ○ 멀티미디어콘텐츠제작전문가

나. 직업활동 영역

- 소프트웨어 개발업체 ○ IT컨설팅 회사 ○ IT교육기관
- 시스템 통합 및 구축 업체 ○ 기업전산실

다. 국가직무능력표준 관련 직종

- SW아키텍처 ○ 응용SW엔지니어링 ○ DB엔지니어링
- NW엔지니어링 ○ 보안엔지니어링 ○ UI/UX엔지니어링
- 임베디드SW엔지니어링

라. 관련 홈페이지 안내

- 훈련기준 및 국가직무능력표준 : <http://www.ncs.go.kr>
- 자격정보 : <http://www.q-net.or.kr>
- 훈련교재 및 매체 : <http://book.hrdkorea.or.kr>

3 출 제 기 준

□ 개발목적

- 각종 자격의 시험문제 작성시 활용하는 기준을 국가직무능력표준에 따라 제시하기 위하여 출제기준(시안)* 개발

* 출제기준(시안) : 출제기준의 경우에는 이를 확정하는 절차를 법령으로 정하여 운영함에 따라 확정된 ‘출제기준’과 국가직무능력표준을 근거로 마련된 출제기준을 구분하기 위하여 ‘출제기준(시안)’ 용어 사용

□ 활용대상

- 국가기술자격법에 따른 국가기술자격
- 개별법령에 따른 국가전문자격
- 자격기본법에 따른 공인민간자격, 민간자격
- 고용보험법에 따른 사업내 자격

□ 활용(예시)

- 자격 및 자격취득자 특성에 따라 능력단위별 출제기준(시안)을 조합하여 출제기준으로 활용

〈방법〉 국가직무능력표준 개발시 관련자격 개선 의견(예시)로 제시된 내용을 그대로 활용

자 격 종 목	능 력 단 위		수 준
	분 류 번 호	명 칭	
궤도기능사(가칭)	14220603_12v1	궤도부설	5
	14220602_12v1	레일용접	4
	14220605_12v1	부대공사	3

1.1. 출제기준(시안)

I. 자격개요

1. 자격 정의

대 분 류	20.정보통신	중 분 류	01.정보기술	소 분 류	02.정보기술개발
자격종목명	응용SW엔지니어링			분류번호	20010202
자격종목정의	컴퓨터 프로그래밍 언어로 각 업무에 맞는 소프트웨어의 기능에 관한 설계, 구현 및 테스트를 수행하고, 사용자에게 배포하며, 버전관리를 통해 제품의 성능을 향상시키고, 서비스를 개선하는 능력이다.				

II. 능력단위별 출제기준(시안)

능 력 단 위		요구사항 확인		능력단위 수준		5수준	
분 류 번 호		2001020201_16v3					
능력단위 정의		업무 분석가가 수집·분석·정의한 요구사항과 이에 따른 분석모델에 대해서 확인과 현행 시스템에 대한 분석하는 능력이다.					
평 가 방 법		지필평가 : 복합형		시 간		60분	
		실무평가 : 수행평가		시 간		60분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020201_16v3.1 현행 시스템 분석하기		1.1 개발하고자 하는 응용소프트웨어에 대한 이해를 높이기 위해, 현행 시스템의 적용현황을 파악함으로써 개발범위와 향후 개발될 시스템으로의 이행방향성을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 운영체제, 데이터베이스관리시스템, 미들웨어 등의 요구사항을 식별할 수 있다. 1.3 현행 시스템을 분석하여, 개발하고자 하는 응용소프트웨어가 이후 적용될 목표시스템을 명확하고 구체적으로 기술할 수 있다.				
	2001020201_16v3.2 요구사항 확인하기		2.1 소프트웨어 공학기술의 요구사항 분석 기법을 활용하여 업무 분석가가 정의한 응용소프트웨어의 요구사항을 확인할 수 있다. 2.2 업무 분석가가 분석한 요구사항에 대해 정의된 검증기준과 절차에 따라서 요구사항을 확인할 수 있다. 2.3 업무 분석가가 수집하고 분석한 요구사항이 개발하고자 하는 응용소프트웨어에 미칠 영향에 대해서 검토하고 확인할 수 있다.				
	2001020201_16v3.3 분석모델 확인하기		3.1 소프트웨어 공학기술의 요구사항 도출 기법을 활용하여 업무 분석가가 제시한 분석모델에 대해서 확인할 수 있다. 3.2 업무 분석가가 제시한 분석모델이 개발할 응용소프트웨어에 미칠 영향을 검토하여 기술적인 타당성 조사를 할 수 있다. 3.3 업무 분석가가 제시한 분석모델에 대해서 응용소프트웨어를 개발하기 위해 필요한 추가적인 의견을 제시할 수 있다.				
관련 지식		<div>- 가상화 관련 지식</div> <div>- 소프트웨어 개발 방법론</div> <div>- 업무 특성에 대한 이해</div> <div>- 요구공학 방법론</div> <div>- 요구분석기법</div> <div>- 클라우드 컴퓨팅 관련 지식</div> <div>- 타당성 분석기법</div> <div>- 통계학</div> <div>- 프로젝트 환경 및 특수성</div> <div>- 프로젝트 환경 및 특수성</div> <div>- 플랫폼에 따른 기능 및 성능 특성</div> <div>- 해당 산업 분야에 대한 지식</div> <div>- 해당 플랫폼에 대한 지식</div>					
평가 시설· 장비		<div>- 컴퓨터</div> <div>- 문서작성 도구</div>					

능 력 단 위		데이터 입출력 구현	능력단위 수준	5수준
분 류 번 호		2001020205_16v4		
능력단위 정의		데이터 입출력 구현은, 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.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	60분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020205_16v4.1 논리 데이터저장소 확인 하기		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.	
	2001020205_16v4.2 물리 데이터저장소 설계 하기		2.1 논리 데이터저장소 설계를 바탕으로 응용소프트웨어가 사용하는 데이터저장소의 특성을 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.	
	2001020205_16v4.3 데이터 조작 프로시저 작 성하기		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.	
	2001020205_16v4.4 데이터 조작 프로시저 최 적화하기		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.	
관련 지식	- E-R Modeling - SQL(Structure Query Language) - 구현에 요구되는 아키텍처의 이해 - 데이터베이스에 대한 이해 - 설계 산출물의 이해 - 성능 평가 기준에 대한 이해 - 소프트웨어 테스트에 대한 이해 - 수학, 통계 지식 - 응용소프트웨어가 사용하는 데이터저장소에 따른 특성(file, RDBMS, XML Database 등)			

	<ul style="list-style-type: none"> - 자료구조 - 프로그래밍 표준에 대한 이해 - 프로그래밍언어(C, C++, Java-JSP/Servlet 등)와 도구(IDE 등)의 활용방법에 대한 이해
평가 시설· 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성 도구 - SW개발 도구 - DBMS - DB설계 도구

능 력 단 위		통합 구현		능력단위 수준	5수준
분 류 번 호		2001020206_16v4			
능력단위 정의		통합 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하여 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력이다.			
평 가 방 법		지필평가 : 복합형		시 간	60분
		실무평가 : 수행평가		시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020206_16v4.1 연계 데이터 구성하기		1.1 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 데이터 연계 요구사항을 분석할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 데이터를 식별할 수 있다. 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계를 위한 데이터 표준을 설계할 수 있다.		
	2001020206_16v4.2 연계 매카니즘 구성하기		2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 특성을 고려하여 효율적 데이터 송수신 방법을 정의할 수 있다. 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 데이터 연계 요구사항을 고려하여 연계주기를 정의할 수 있다. 2.3 개발하고자하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 데이터 연계 실패 시 처리방안을 정의할 수 있다. 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 송수신 시 보안을 적용할 수 있다.		
	2001020206_16v4.3 내외부 연계 모듈 구현하기		3.1 구성된 연계 매카니즘에 대한 명세서를 참조하여 연계모듈구현을 위한 논리적, 물리적 환경을 준비할 수 있다. 3.2 구성된 연계 매카니즘에 대한 명세서를 참조하여 외부 시스템과의 연계 모듈을 구현할 수 있다. 3.3 연계모듈의 안정적인 작동여부와 모듈 간 인터페이스를 통해 연동된 데이터의 무결성을 검증할 수 있다. 3.4 구현된 연계모듈을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.		
관련 지식		<ul style="list-style-type: none">- 개발 방법론 이해- 구현에 요구되는 아키텍처의 이해- 네트워크 이해- 데이터베이스 이해- 분산 프로그래밍에 대한 이해- 설계 모델링 기법- 설계 산출물의 이해- 소프트웨어 아키텍처에 대한 이해- 소프트웨어 테스트에 대한 이해- 업무 특성에 대한 이해- 프로그래밍어 활용방법의 이해- 해당 산업 분야에 대한 지식			
평가 시설· 장비		<ul style="list-style-type: none">- 컴퓨터- 문서작성 도구- SW개발 도구			

능 력 단 위		정보시스템 이행		능력단위 수준	5수준
분 류 번 호		2001020208_16v3			
능력단위 정의		정보시스템 이행은 개발자 환경에서 개발한 결과물을 운영 환경에 설치하고, 사용자 요구사항과 최종적으로 일치하는 지에 대해 승인을 얻어 응용소프트웨어 결과물을 사용자에게 전달하여 인계하고 시스템을 운영할 수 있도록 교육하고 지원하는 능력이다.			
평 가 방 법		지필평가 : 복합형		시 간	60분
		실무평가 : 시뮬레이션		시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020208_16v3.1 정보시스템 사용자 교육하기		1.1 개발된 응용소프트웨어 사용자 교육을 위해 교육 항목과 대상자, 강사선정, 방법, 시간, 횟수 등을 포함한 교육계획을 수립할 수 있다. 1.2 개발된 응용소프트웨어의 기능, 사용법과 필요 기술에 관한 사항을 포함한 사용자 교육 교재를 개발할 수 있다. 1.3 효과적인 사용자 교육이 이루어질 수 있도록 사전에 필요한 교육환경을 구축하고 점검 할 수 있다. 1.4 교육계획에 따라 사용자 교육을 실시하고, 평가기준에 따라 교육결과 및 성과를 평가할 수 있다.		
	2001020208_16v3.2 정보시스템 이행하기		2.1 하드웨어, 시스템 소프트웨어, 네트워크 등 시스템 운영환경을 설치하고, 개발된 애플리케이션을 운영환경에 설치할 수 있다. 2.2 시스템 운영방법 및 사용법과 필요 기술에 관한 사항을 포함한 운영자 매뉴얼을 개발할 수 있다. 2.3 정보시스템 이행을 위한 각 영역 별 사전점검표를 작성하고, 데이터 이행작업 및 검증할 수 있다. 2.4 정보시스템 이행 시 발생할 수 있는 실패나 사고에 대비하여 신속하게 지원할 수 있는 비상 대책을 수립할 수 있다.		
	2001020208_16v3.3 정보시스템 안정화하기		3.1 정보시스템 이행 후 개발된 응용소프트웨어의 데이터, 오류, 사용자 추가 요구사항을 수집하고 검토할 수 있다. 3.2 정보시스템 이행 후 실행되는 과정 중에 발생할 수 있는 정보시스템의 자원 사용량, 운영 상황을 분석하여 정보시스템을 안정화할 수 있다. 3.3 개발된 응용소프트웨어가 적용되는 정보시스템의 성능을 분석하여 문제점을 파악한 뒤 성능을 개선할 수 있다.		
관련 지식		<div>- 국제 표준 제품 품질특성에 대한 사항</div> <div>- 문서작성 도구 활용 방법</div> <div>- 소프트웨어 아키텍처에 대한 이해</div> <div>- 시스템 성능 분석 및 진단에 대한 지식</div> <div>- 시스템 운영 환경에 대한 이해</div> <div>- 시스템 운영에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해</div> <div>- 업무 특성에 대한 이해</div> <div>- 표준 양식(Template) 작성 및 활용법</div> <div>- 피교육 집단 특성에 대한 이해</div> <div>- 해당 산업 분야에 대한 지식</div>			
평가 시설· 장비		<div>- 컴퓨터</div> <div>- 문서작성 도구</div> <div>- SW개발 도구</div>			

능 력 단 위	제품소프트웨어 패키징	능력단위 수준	5수준
분 류 번 호	2001020209_16v4		
능력단위 정의	제품소프트웨어 패키징은 개발이 완료된 제품소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 제품소프트웨어에 대한 설치와 사용에 필요한 제반 절차 및 환경 등 제품소프트웨어의 전체 내용을 포함하는 매뉴얼을 작성하며, 제품소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전관리를 수행하는 능력이다.		
평 가 방 법	지필평가 : 복합형	시 간	60분
	실무평가 : 시뮬레이션	시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)	수 행 준 거 (세 세 항 목)	
	2001020209_16v4.1 제품소프트웨어 패키징하기	1.1 신규 개발, 변경, 개선된 제품소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징 할 수 있다. 1.2 이전 릴리즈 이후의 변경, 개선사항을 포함하여 신규 패키징한 제품소프트웨어에 대한 릴리즈 노트를 작성할 수 있다. 1.3 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품소프트웨어의 설치, 배포 파일을 생성할 수 있다.	
	2001020209_16v4.2 제품소프트웨어 매뉴얼 작성하기	2.1 사용자가 제품소프트웨어를 설치하는데 참조할 수 있도록 제품소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. 2.2 사용자가 제품소프트웨어를 사용하는데 참조할 수 있도록 제품소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다. 2.3 사용자가 제품소프트웨어를 설치하고 사용하는데 필요한 제품소프트웨어의 설치파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.	
	2001020209_16v4.3 제품소프트웨어 버전관리하기	3.1 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다. 3.2 형상관리 지침을 활용하여 제품소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다. 3.3 버전 관리 도구를 활용하여 제품소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.	
관련 지식	- 소프트웨어 버전관리 및 도구 사용에 대한 이해 - 장애 복구 처리 절차 - 각종 매뉴얼 작성법 - 국제 표준 제품 품질특성에 대한 사항 - 릴리즈 노트 작성법 - 사용자의 제품 실행 환경에 대한 이해 - 제품 패키징, 배포 관련 표준에 대한 지식 - 제품소프트웨어 관련 시장에 대한 지식 - 제품소프트웨어 사용에 대한 데이터(오류, 자원사용량 등) 수집 및 분석에 대한 이해 - 제품소프트웨어 아키텍처에 대한 이해 - 제품소프트웨어가 적용되는 업무 특성에 대한 이해		
평가 시설·장비	- 컴퓨터 - 문서작성 도구 - SW개발 도구 - SW패키징 도구		

능 력 단 위	서버프로그램 구현	능력단위 수준	5수준
분 류 번 호	2001020211_16v4		
능력단위 정의	서버프로그램 구현이란 애플리케이션 설계를 기반으로 개발에 필요한 환경을 구성하고, 프로그래밍 언어와 도구를 활용하여 공통모듈, 업무프로그램과 배치 프로그램을 구현하는 능력이다.		
평 가 방 법	지필평가 : 단답형	시 간	60분
	실무평가 : 수행평가	시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)	수 행 준 거 (세 세 항 목)	
	2001020211_16v4.1 개발환경 구축하기	1.1 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어의 필요 사항을 검토하고 이에 따라, 개발환경에 필요한 준비를 수행할 수 있다. 1.2 응용소프트웨어 개발에 필요한 하드웨어 및 소프트웨어를 설치하고 설정하여 개발환경을 구축할 수 있다. 1.3 사전에 수립된 형상관리 방침에 따라, 운영정책에 부합하는 형상관리 환경을 구축할 수 있다.	
	2001020211_16v4.2 공통 모듈 구현하기	2.1 공통 모듈의 상세 설계를 기반으로 프로그래밍 언어와 도구를 활용하여 업무 프로세스 및 서비스의 구현에 필요한 공통 모듈을 작성할 수 있다. 2.2 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도를 높인 공통모듈을 구현할 수 있다. 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.	
	2001020211_16v4.3 서버 프로그램 구현하기	3.1 업무 프로세스 맵과 세부 업무 프로세스를 확인할 수 있다. 3.2 세부 업무프로세스를 기반으로 프로그래밍 언어와 도구를 활용하여 서비스의 구현에 필요한 업무 프로그램을 구현할 수 있다. 3.3 개발하고자 하는 목표 시스템의 잠재적 보안 취약성이 제거될 수 있도록 서버 프로그램을 구현할 수 있다. 3.4 개발된 업무 프로그램의 내부 기능과 제공하는 인터페이스에 대해 테스트를 수행할 수 있다.	
	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.	
관련 지식	- 개발에 필요한 하드웨어 및 소프트웨어 사양 - 개발환경에 요구되는 아키텍처		

	<ul style="list-style-type: none"> - 서버 개발 프레임워크 개념 - 설계 산출물 해석 방법 - 데이터베이스 개념 - 네트워크 개념 - 보안 관련 개념 - 개발환경에 요구되는 아키텍처 - 설계 산출물 해석 방법 - 프로그램 언어 활용방법 - 데이터베이스 개념 - 네트워크 개념 - 분산 처리 개념 - 소프트웨어 테스트 - 구현에 요구되는 아키텍처 - 설계 산출물 해석방법 - 프로그램 언어 활용방법 - 단위 업무 - 단위테스트 기술동향 및 도구 - 소프트웨어 테스트 - 결함 관리 - 구현에 요구되는 아키텍처 - 설계 산출물 해석방법 - 프로그램 언어 활용방법 - 데이터베이스 개념 - 네트워크 개념 - 분산 프로그래밍 개념 - 소프트웨어 테스트
평가 시설 · 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성도구 - SW개발도구

능 력 단 위		인터페이스 구현	능력단위 수준	5수준
분 류 번 호		2001020212_16v4		
능력단위 정의		인터페이스 구현이란 모듈간의 분산이 이루어진 경우를 포함하여 단위 모듈간의 데이터 관계를 분석하고 이를 기반으로 한 메커니즘을 통해 모듈간의 효율적인 연계를 구현하고 검증하는 능력이다.		
평 가 방 법		지필평가 : 단답형	시 간	60분
		실무평가 : 수행평가	시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)	수 행 준 거 (세 세 항 목)		
	2001020212_16v4.1 인터페이스 설계서 확인하기	1.1 인터페이스 설계서를 기반으로 외부 및 내부 모듈 간의 공통적으로 제공되는 기능과 각 데이터의 인터페이스를 확인할 수 있다. 1.2 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 연계가 필요한 인터페이스의 기능을 식별할 수 있다. 1.3 개발하고자 하는 응용소프트웨어와 관련된 외부 및 내부 모듈 간의 인터페이스를 위한 데이터 표준을 확인할 수 있다.		
	2001020212_16v4.2 인터페이스 기능 구현하기	2.1 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 일관되고 정형화된 인터페이스 기능 구현을 정의할 수 있다. 2.2 개발하고자 하는 응용소프트웨어와 연계 대상 모듈 간의 세부 설계서를 확인하여 공통적인 인터페이스를 구현할 수 있다. 2.3 개발하고자하는 응용소프트웨어와 연계 대상 내외부 모듈 간의 연계 목적을 고려하여 인터페이스 기능 구현 실패 시 예외처리방안을 정의할 수 있다. 2.4 응용소프트웨어와 관련된 내외부 모듈 간의 연계 데이터의 중요성을 고려하여 인터페이스 보안 기능을 적용할 수 있다.		
	2001020212_16v4.3 인터페이스 구현 검증하기	3.1 구현된 인터페이스 명세서를 참조하여 구현 검증에 필요한 감시 및 도구를 준비할 수 있다. 3.2 인터페이스 구현 검증을 위하여 외부 시스템과의 연계 모듈 상태를 확인할 수 있다. 3.3 인터페이스 오류처리 사항을 확인하고 보고서를 작성할 수 있다.		
관련 지식	- IDE 및 개발환경 도구 활용 - UML 작성 기술 - 개발 방법론 - 구현에 요구되는 아키텍처 - 기술영역별 미들웨어/솔루션 활용 - 네트워크 - 데이터 보안 기술 - 데이터 연계/이관 도구 활용 능력			

	<ul style="list-style-type: none"> - 데이터베이스 - 디자인 패턴 - 분산 프로그래밍 - 산업 분야별 업무특성 - 설계 모델링 기법 - 설계 모델링 기술 - 설계 산출물 - 소프트웨어 아키텍처 - 소프트웨어 테스트 - 프레임워크 활용
평가 시설· 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성도구 - SW개발도구

능 력 단 위		애플리케이션 배포		능력단위 수준		4 수준	
분 류 번 호		2001020214_15v3					
능력단위 정의		애플리케이션 배포란 애플리케이션 배포 환경을 구성하고, 구현이 완료된 애플리케이션의 소스 검증 및 빌드를 수행하여 운영 환경에 배포하는 능력이다.					
평 가 방 법		지필평가 : 단답형		시 간		60분	
		실무평가 : 수행평가		시 간		90분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020214_15v3.1 애플리케이션 배포 환경 구성하기		1.1 애플리케이션 빌드와 배포를 위한 환경 구성 방안을 계획할 수 있다. 1.2 애플리케이션 배포를 위한 도구와 시스템을 결정할 수 있다. 1.3 결정한 애플리케이션 배포 환경을 위한 도구와 시스템을 설치할 수 있다. 1.4 설치한 시스템과 도구 운영을 위해 상세 구성 및 설정을 할 수 있다.				
	2001020214_15v3.2 애플리케이션 소스 검증 하기		2.1 정상적으로 작동하는 소프트웨어 빌드를 위해 형상관리 서버로부터 소스코드를 체크 아웃 할 수 있다. 2.2 소스코드 검증 도구를 활용하여 애플리케이션에서 사용한 라이브러리, 소스, 로직 등의 오류가 있는지 여부를 검증할 수 있다. 2.3 소스 코드의 환경 설정, 운영 환경 정보, 대상 시스템 정보 등에 오류가 있는지 확인할 수 있다.				
	2001020214_15v3.3 애플리케이션 빌드하기		3.1 애플리케이션 소스코드 검증 결과 문제가 없는 경우 해당 소스코드를 빌드 시스템으로 이관할 수 있다. 3.2 애플리케이션 빌드 절차에 따른 빌드 스크립트를 작성할 수 있다. 3.3 작성한 빌드 스크립트 또는 도구를 활용하여 애플리케이션 빌드를 실행할 수 있다. 3.4 애플리케이션 빌드 실행 결과를 확인하여 정상적으로 완료되었는지 여부를 확인할 수 있다. 3.5 애플리케이션 빌드 실패 시 문제 내용과 원인을 파악하여 개발자에게 설명할 수 있다.				
	2001020214_15v3.4 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.				
관련 지식	- 빌드 결과 분석 지식 - 소스코드 검증 기법						

	<ul style="list-style-type: none"> - 소스코드 검증 기준 - 소스코드 검증 도구 - 소스코드 동적 검증 분석 기법 - 소스코드 정적 검증 분석 기법 - 애플리케이션 빌드 스크립트 문법 - 애플리케이션 운영 플랫폼 정보 - 애플리케이션 운영 환경 배포 절차 - 애플리케이션 원복 절차 - 형상관리 지침 - 배포 도구 설치 및 사용법 - 애플리케이션 배포 환경 구성 지식 - 형상관리 도구 설치 및 사용법 - 애플리케이션 빌드(Build) 절차 - 형상관리 도구 사용법 - 애플리케이션 빌드 시스템 가이드
평가 시설· 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성도구 - SW개발도구

능 력 단 위		프로그래밍 언어 활용	능력단위 수준		3 수준
분 류 번 호		2001020215_15v3			
능력단위 정의		프로그래밍 언어 활용이란 응용소프트웨어 개발에 사용되는 프로그래밍 언어의 기초 문법을 적용하고 언어의 특징과 라이브러리를 활용하여 기본 응용소프트웨어를 구현하는 능력이다.			
평 가 방 법		지필평가 : 단답형	시 간		60분
		실무평가 : 수행평가	시 간		90분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020215_15v3.1 기본문법 활용하기		1.1 응용소프트웨어 개발에 필요한 프로그래밍 언어의 데이터 타입을 적용하여 변수를 사용할 수 있다. 1.2 프로그래밍 언어의 연산자와 명령문을 사용하여 애플리케이션에 필요한 기능을 정의하고 사용할 수 있다. 1.3 프로그래밍 언어의 사용자 정의 자료형을 정의하고 애플리케이션에서 사용할 수 있다.		
	2001020215_15v3.2 언어특성 활용하기		2.1 프로그래밍 언어별 특성을 파악하고 설명할 수 있다. 2.2 파악된 프로그래밍 언어의 특성을 적용하여 애플리케이션을 구현할 수 있다. 2.3. 애플리케이션을 최적화하기 위해 프로그래밍 언어의 특성을 활용 할 수 있다.		
	2001020215_15v3.3 라이브러리 활용하기		3.1 애플리케이션에 필요한 라이브러리를 검색하고 선택할 수 있다. 3.2 애플리케이션 구현을 위해 선택한 라이브러리를 프로그래밍 언어 특성에 맞게 구성 할 수 있다. 3.3 선택한 라이브러리를 사용하여 애플리케이션 구현에 적용할 수 있다.		
관련 지식	<ul style="list-style-type: none">- 프로그래밍 언어 기본 문법- 프로그래밍 언어 문법 오류- 프로그래밍 언어 기능 구현을 위한 알고리즘- 디버깅 기법- 코드 검토 기법- 절차적 프로그래밍 언어- 객체지향 프로그래밍 언어- 스크립트 언어- 선언형 언어- 라이브러리 기능에 대한 이해- 애플리케이션 요구사항에 대한 이해- 라이브러리 적용을 위한 기술(파일입출력, 데이터입출력, 예외 처리 등)에 대한 이해				
평가 시설· 장비	<ul style="list-style-type: none">- 컴퓨터- 문서작성도구- SW개발도구				

능 력 단 위	응용 SW 기초 기술 활용	능력단위 수준	3 수준
분 류 번 호	2001020216_15v3		
능력단위 정의	응용 SW 기초 기술 활용이란 응용소프트웨어개발을 위하여 운영체제, 데이터베이스, 네트워크의 기초 기술을 적용하고 응용개발에 필요한 환경을 구축하는 능력이다.		
평 가 방 법	지필평가 : 단답형	시 간	60분
	실무평가 : 수행평가	시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)	수 행 준 거 (세 세 항 목)	
	2001020216_15v3.1 운영체제 기초 활용하기	1.1 응용 소프트웨어를 개발하기 위하여 다양한 운영체제의 특징을 설명할 수 있다. 1.2 CLI(Command Line Interface) 및 GUI(Graphic User Interface) 환경에서 운영체제의 기본명령어를 활용할 수 있다. 1.3 운영체제에서 제공하는 작업 우선순위 설정방법을 이용하여 애플리케이션의 작업우선순위를 조정할 수 있다.	
	2001020216_15v3.2 데이터베이스 기초 활용하기	2.1 데이터베이스의 종류를 구분하고 응용 소프트웨어 개발에 필요한 데이터베이스를 선정할 수 있다. 2.2 주어진 E-R 다이어그램을 이용하여 관계형 데이터베이스의 테이블을 정의할 수 있다. 2.3 데이터베이스의 기본연산을 CRUD(Create, Read, Update, Delete)로 구분하여 설명할 수 있다.	
	2001020216_15v3.3 네트워크 기초 활용하기	3.1 네트워크 계층구조에서 각 층의 역할을 설명할 수 있다. 3.2 응용의 특성에 따라 TCP와 UDP를 구별하여 적용할 수 있다. 3.3 패킷 스위칭 시스템을 이해하고, 다양한 라우팅 알고리즘과 IP 프로토콜을 설명할 수 있다.	
	2001020216_15v3.4 기본 개발환경 구축하기	4.1 응용개발을 위하여 선정된 운영체제를 설치하고 운용할 수 있다. 4.2 응용개발에 필요한 개발도구를 설치하고 운용할 수 있다. 4.3 웹서버, DB서버 등 응용개발에 필요한 기반 서버를 설치하고 운용할 수 있다.	
관련 지식	- 각 운영체제 특징에 대한 이해 - Windows 운영체제 기본 명령어 - 리눅스 / 유닉스 계열 운영체제 기본 명령어 - 메모리 관리 기법의 이해 - 프로세스 스케줄링 기법의 이해 - 운영체제 환경변수 설정 방법 - ER 다이어그램 작성 방법 - 무결성 제약조건의 이해 - 테이블 선언 및 조작용어 - 릴레이션의 특징 - DBMS 각 유형별 특징 - CRUD 연산에 대한 이해		

	<ul style="list-style-type: none"> - 인터넷 구성의 개념 - 프로토콜 개념 - IP 주소 체계 - 트랜스포트 방식 개념 - TCP/UDP 방식 개념 - 네트워크 7 계층의 개념 - 버전관리 시스템 구축 - 운영체제 설치 및 제거 방법 - 개발 프로그램 설치 방법 - 라이브러리 및 필요 패키지 설치 방법 - 필요 응용 소프트웨어에 대한 이해 - 사용자의 프로그램 실행 환경에 대한 이해
평가 시설· 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성도구 - SW개발도구

능 력 단 위		애플리케이션 리팩토링		능력단위 수준		6수준	
분 류 번 호		2001020217_16v4					
능력단위 정의		애플리케이션 리팩토링이란 소스코드가 수행하는 기능을 유지하면서 코드 가독성 및 잠재적 결함을 제거하도록 코드의 구조를 개선하는 능력이다.					
평 가 방 법		지필평가 : 복합형		시 간		60분	
		실무평가 : 수행평가		시 간		120분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020217_16v4.1 리팩토링 기준 수립하기		1.1 리팩토링을 적용하여 달성하려는 품질 목표를 수립할 수 있다. 1.2 리팩토링 결과의 완전성을 검증할 수 있는 체크리스트를 수립할 수 있다. 1.3 리팩토링에 활용할 수 있는 도구 및 절차를 수립할 수 있다.				
	2001020217_16v4.2 소스 코드 분석하기		2.1 개발 표준을 만족하지 않는 소스 코드를 식별할 수 있다. 2.2 성능개선과 이해도 증진을 위하여 불필요한 소스 코드를 식별할 수 있다. 2.3 잠재 결함 요소를 내포한 소드 코드를 식별할 수 있다.				
	2001020217_16v4.3 리팩토링 적용하기		3.1 리팩토링을 위한 소스 코드 구조를 개선할 수 있다. 3.2 개선한 소스 코드가 초기 기능을 유지하는지 확인할 수 있다. 3.3 개선한 소스 코드가 리팩토링 체크리스트를 만족하는지 검토할 수 있다.				
관련 지식		- 국제 코딩 표준 - 디자인 패턴 - 리팩토링 기법 - 리팩토링 적용 기법 - 비기능적 품질 특성 - 비기능적 품질 특성 충족 기법 - 코드 스멜 유형 - 클린 코드 특징 - 클린 코드의 필요성 및 특징 - 회귀 테스트 - 비기능적 품질 특성					
평가 시설· 장비		- 컴퓨터 - 문서작성도구 - SW개발도구					

능 력 단 위		인터페이스 설계		능력단위 수준	6수준
분 류 번 호		2001020218_16v4			
능력단위 정의		인터페이스 설계란 응용소프트웨어 개발을 위해 정의된 시스템 인터페이스 요구사항을 확인하고 인터페이스 대상을 식별하여 인터페이스 설계서를 작성하는 능력이다.			
평 가 방 법		지필평가 : 복합형		시 간	60분
		실무평가 : 시뮬레이션		시 간	90분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020218_16v4.1 인터페이스 요구사항 확 인하기		1.1 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 비 기능 요구사항을 분석할 수 있다. 1.2 정의 된 응용소프트웨어 요구사항을 참조하여 개발하고자 하는 응용소프트웨어의 인터페이스 기능 요구사항을 분석할 수 있다. 1.3 개발하고자 하는 응용소프트웨어의 인터페이스 요구사항의 정확성과 완전성을 확인한다.		
	2001020218_16v4.2 인터페이스 대상 식별하 기		2.1 개발하고자 하는 응용소프트웨어의 내부와 외부 인터페이스 대상 시스템을 식별할 수 있다. 2.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 연계 시스템을 식별할 수 있다. 2.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 송수신 데이터를 식별할 수 있다.		
	2001020218_16v4.3 인터페이스 상세 설계하 기		3.1 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위한 송수신 방법을 명세화할 수 있다. 3.2 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스를 위해 필요한 데이터를 명세화할 수 있다. 3.3 개발하고자하는 응용소프트웨어의 내부와 외부 인터페이스의 오류 시 처리방안을 명세화할 수 있다. 3.4 소프트웨어 아키텍처에서 정의한 인터페이스 설계 기준에 따라 외부와 내부 시스템 간의 인터페이스를 설계서를 작성할 수 있다.		
관련 지식	- 네트워크 - 산업 분야별 업무특성 - 설계 모델링 방법론 - 소프트웨어 아키텍처 - 시스템 아키텍처 - 요구공학 - 인터페이스 시스템				
평가 시설· 장비	- 컴퓨터 - 문서작성도구 - SW개발도구				

능 력 단 위		애플리케이션 요구사항 분석		능력단위 수준	7 수준
분 류 번 호		2001020219_16v4			
능력단위 정의		애플리케이션 요구사항 분석이란 구현하고자 하는 애플리케이션의 요구사항을 도출, 분석, 명세화 및 요구사항 검증을 수행하는 능력이다.			
평 가 방 법		지필평가: 복합형		시 간	60분
		실무평가: 수행평가		시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020219_16v4.1 요구사항 도출하기		1.1 비즈니스 목적, 도메인 지식, 이해당사자, 비즈니스 규칙, 조직 환경 등 다양한 요구사항 도출을 위한 정보를 수집 할 수 있다. 요구사항 도출을 위한 정보를 수집 할 수 있다. 1.2 인터뷰, 시나리오 작성, 프로토타입 제작, 그룹회의, 관찰, 사용자 스토리 작성 등 제시된 기법을 통해 어플리케이션의 요구사항을 도출 할 수 있다. 1.3 요구사상 추출 내용을 관리하기 위해 요구사항의 출처와 요구 내용을 상세하게 기록할 수 있다.		
	2001020219_16v4.2 요구사항 분석하기		2.1 요구사항 분석과 협상을 위하여 도출한 요구사항을 기능적 요구사항과 비기능적 요구사항을 분류 할 수 있다. 2.2 도출한 요구사항의 내용을 분석하여 구현을 위한 우선순위를 결정할 수 있다. 2.3 도출한 요구사항이 개발하고자 하는 애플리케이션 전체 또는 단위 컴포넌트에 향에 영향을 미치는지 여부를 분석하여 요구사항의 적용 범위를 확인할 수 있다. 2.4 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.		
	2001020219_16v4.3 요구사항 명세화하기		3.1 애플리케이션 사용자 또는 활용 분야의 관점에서 요구사항을 분석한 결과를 정리한 상위 수준의 사용자 요구사항 정의서를 작성할 수 있다. 3.2 구현하고자 하는 애플리케이션이 구동되는 시스템이나 하드웨어 관련 내용을 구체화 하여 시스템 요구사항 문서를 작성할 수 있다. 3.3 구현하고자 하는 애플리케이션에 대해 설명, 검증 및 테스트, 계약을 위해 사용자 요구사항 정의서와 시스템 요구사항 문서를 토대로 요구사항을 구체화하여 소프트웨어 요구사항 명세서(SRS, Software Requirement Specification)을 작성할 수 있다.		
	2001020219_16v4.4 요구사항 검증하기		4.1 소프트웨어 요구사항 명세서가 문서 표준을 준수하여 사용자가 이해 가능한 내용으로 일관성 있고, 완성도 있게 작성되었는지를 검증할 수 있다. 4.2 작성한 요구사항 명세서의 오류, 잘못된 가정, 불명확한		

		<p>설명, 비표준적인 내용 등이 있는지 확인을 위해 조사(Inspection) 혹은 검토(Review)를 실시할 수 있다.</p> <p>4.3 기술 환경, 구현 가능성, 안정성 등을 고려하여 요구된 도출 사항들을 검증할 수 있다.</p> <p>4.4 분석 단계의 개념 모델링 결과물이 있을 경우 표기법, 모델 원칙 준수 여부 판단 등을 통해 모델을 검증할 수 있다.</p>
관련 지식	<ul style="list-style-type: none"> - SW 제품 개발 수명 주기 이론 - 요구 공학(Requirement Engineering) 이론 - SW 제품 요구사항 관리 절차 - SW 제품 요구사항 도출 지식 - SW 제품 요구사항 정의 기준 - 요구사항 명세서(Software Requirement Specification) 작성 기준 - IEEE830 (IEEE Recommended Practice for Software Requirements Specifications) - 정보시스템마스터플랜 ISMP(Information System Master Plan) 방법론 - BABOK (Business Analysis Body of Knowledge) - SWEBOOK(Software Engineering Body of Knowledge) - SW제품 개발 요구 공학 - SW제품 품질 속성 항목 - SW제품 품질 평가 항목 - 정보시스템의 기술적 지식 - 요구사항 적용 범위 확인 지식 - 요구사항 우선 순위 결정 기준 - 요구사항 영향범위 설정 기준 	
평가 시설· 장비	<ul style="list-style-type: none"> - 컴퓨터 - 문서작성도구 - SW개발도구 	

능 력 단 위		기능모델 설계	능력단위 수준	6 수준
분 류 번 호		2001020220_16v1		
능력단위 정의		기능모델 설계란 애플리케이션 요구사항에 의해 도출, 분석된 내용을 애플리케이션의 기능으로 명세화하여 요구사항을 검증하고 소프트웨어 개발범위를 확정하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020220_16v1.1 기능모델 작성하기		1.1 기능모델을 작성하기 위해 필요한 다이어그램을 선정할 수 있다. 1.2 기능모델링을 위해 선정한 다이어그램의 각 구성요소를 식별하여 다이어그램을 작성할 수 있다. 1.3 기능모델링을 위해 선정한 다이어그램을 설명하기 위한 기능모델 설계 명세서를 작성할 수 있다.	
	2001020220_16v1.2 기능모델 검증하기		2.1 도식된 기능모델이 표준 표기법을 준수하고 있는지 검증할 수 있다. 2.2 작성된 기능모델이 표준으로 제시된 산출물 양식과 지침을 따르고 있는지 검증할 수 있다. 2.3 작성된 기능모델이 정의된 소프트웨어 요구사항을 모두 포함하고 있는지 검증할 수 있다. 2.4 요구사항 분석 결과를 토대로 사용자와 협상을 통해 요구사항, 우선순위, 영향 범위 등을 확정할 수 있다.	
관련 지식	- 해당 산업 분야에 대한 지식 - 업무 특성에 대한 이해 - 개발 방법론 이해 - 요구 공학(Requirement Engineering) 이론 - SWEBOK(Software Engineering Body of Knowledge)			
평가 시설· 장비	- 컴퓨터 - 문서작성도구 - SW개발도구			

능 력 단 위		애플리케이션 설계		능력단위 수준		6 수준	
분 류 번 호		2001020221_16v4					
능력단위 정의		애플리케이션 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 이에 따른 애플리케이션 구현을 수행하기 위해 공통모듈 설계, 타 시스템 연동에 대하여 상세 설계하는 능력이다.					
평 가 방 법		지필평가 : 복합형		시 간		60분	
		실무평가 : 수행평가		시 간		120분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020221_16v4.1 공통 모듈 설계하기		1.1 재사용성 확보와 중복개발을 회피하기 위하여, 전체 시스템 차원과 단위 시스템 차원의 공통 부분을 식별하여 이에 대한 상세 명세를 작성할 수 있다. 1.2 개발할 응용소프트웨어의 전반적인 기능과 구조를 이해하기 쉬운 크기로 공통 모듈을 설계할 수 있다. 1.3 소프트웨어 측정지표 중 모듈간의 결합도는 줄이고 개별 모듈들의 내부 응집도는 높이기 위한 공통모듈을 설계할 수 있다. 1.4 전반적인 처리 논리 구조에 예기치 못한 영향을 끼치지 않도록 공통 모듈 인터페이스의 인덱스 번호나 기능 코드를 설계할 수 있다.				
	2001020221_16v4.2 타 시스템 연동설계하기		2.1 소프트웨어 아키텍처에서 정의한 타 시스템 연동 리스트 및 연동 방안을 참조하여, 타 시스템 연동 상세 설계의 가이드라인을 작성할 수 있다. 2.2 소프트웨어 아키텍처의 정의를 반영한 연동 상세 설계 가이드라인에 따라, 타 시스템 연동 상세 설계할 수 있다. 2.3 소프트웨어 아키텍처에 따라 선정된 개발 및 운영 환경에 사용될 기술영역별 미들웨어/솔루션에 대하여 명세를 작성할 수 있다. 2.4 소프트웨어 아키텍처에 따른 시스템간의 연동 시, 발생할 수 있는 오류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.				
	관련 지식		- 해당 산업 분야에 대한 지식 - 업무 특성에 대한 이해 - 개발 방법론 이해 - 소프트웨어 아키텍처에 대한 이해 - 설계 모델링 기법 - 네트워크 이해 - 분산 프로그래밍에 대한 이해				
평가 시설· 장비		- 컴퓨터 - 문서작성도구 - SW개발도구					

능 력 단 위		정적모델 설계	능력단위 수준	6 수준
분 류 번 호		2001020222_16v4		
능력단위 정의		정적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드 라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 정적모 델을 설계하고 검증하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)	수 행 준 거 (세 세 항 목)		
	2001020222_16v4.1 정적 분석모델 검증하기	1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 정적모델 상세분석 내 역을 확인할 수 있다. 1.2 분석된 정적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.		
	2001020222_16v4.2 정적모델 상세화하기	2.1 분석된 정적모델을 정적 설계모델로 변환하기 위해 변환규칙을 정의할 수 있다. 2.2 정적 설계모델의 엔티티 클래스, 경계 클래스, 제어 클래스를 상세화할 수 있다. 2.3 상세화된 정적 설계모델의 명세서를 작성할 수 있다. 2.4 설계용 도구를 활용하여 정적 설계모델을 도식할 수 있다.류를 예측하고 이의 대응 방안에 대해 제시할 수 있다.		
관련 지식	- 해당 산업 분야에 대한 지식 - 업무 특성에 대한 이해 - 개발 방법론 이해 - 소프트웨어 아키텍처에 대한 이해 - 설계 모델링 기법 - 네트워크 이해 - 데이터베이스 이해			
평가 시설· 장비	- 컴퓨터 - 문서작성도구 - SW개발도구			

능 력 단 위		동적모델 설계	능력단위 수준	6 수준
분 류 번 호		2001020223_16v4		
능력단위 정의		동적모델 설계란 요구사항 확인을 통한 상세 분석 결과, 소프트웨어 아키텍처 가이드라인 및 소프트웨어 아키텍처 산출물에 의거하여 애플리케이션 구현을 위한 동적모델을 설계하고 검증하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020223_16v4.1 동적 분석모델 검증하기		1.1 소프트웨어 아키텍처 설계 가이드라인을 참조하여 동적모델 상세분석 내역을 확인할 수 있다. 1.2 분석된 동적모델을 대상으로 분석모델의 정합성을 검증할 수 있다.	
	2001020223_16v4.2 동적모델 상세화 하기		2.1 검증된 동적 분석모델을 동적 설계모델로 상세화하기 위한 적절한 다이어그램을 선정할 수 있다. 2.2 동적 설계모델을 선정한 다이어그램을 이용하여 상세 설계할 수 있다. 2.3 상세 설계된 동적 모델을 설계용 도구를 활용하여 도식화할 수 있다. 2.4 도식화한 동적 설계모델을 통해 기능모델과 정적모델을 상호 검증하고 조정할 수 있다.	
관련 지식		- 해당 산업 분야에 대한 지식 - 업무 특성에 대한 이해 - 개발 방법론 이해 - 소프트웨어 아키텍처에 대한 이해 - 설계 모델링 기법 - 네트워크 이해 - 분산 프로그래밍에 대한 이해		
평가 시설· 장비		- 컴퓨터 - 문서작성도구 - SW개발도구		

능 력 단 위		화면 설계	능력단위 수준	5 수준
분 류 번 호		2001020224_16v4		
능력단위 정의		화면 설계란 요구사항분석 단계에서 파악된 화면에 대한 요구사항을 소프트웨어 아키텍처 단계에서 정의된 구현 지침 및 UI/UX 엔지니어가 제시한 UI표준과 지침에 따라 화면을 설계하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020224_16v4.1 UI 요구사항 확인하기		1.1 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, 개발하고자 하는 응용소프트웨어에 적용될 UI 요구사항을 확인할 수 있다. 1.2 응용소프트웨어 개발을 위한 UI 표준 및 지침에 의거하여, UI 요구사항을 반영한 프로토타입을 제작할 수 있다. 1.3 작성한 프로토타입을 활용하여 UI/UX엔지니어와 향후 적용할 UI의 적정성에 대해 검토할 수 있다.	
	2001020224_16v4.2 UI 설계하기		2.1 UI 요구사항과 UI 표준 및 지침에 따라, 화면과 품의 흐름을 설계하고, 제약사항을 화면과 품 흐름 설계에 반영할 수 있다. 2.2 UI 요구사항과 UI 표준 및 지침에 따라, 사용자의 편의성을 고려한 메뉴 구조를 설계할 수 있다. 2.3 UI 요구사항과 UI 표준 및 지침에 따라, 하위 시스템 단위의 내·외부 화면과 품을 설계할 수 있다.	
	관련 지식		- 구현에 요구되는 아키텍처의 이해 - 유용성 개념 및 적용원리 - UI 기획/설계/구현 절차 - UI 표준 및 지침 - 유즈케이스에 대한 이해 - 데이터 타입 특성 - 사용자 기반 메뉴 구조 작성 방법 - 감성공학에 대한 이해 - 프로토타입 작성법	
평가 시설· 장비		- 컴퓨터 - 문서작성도구 - SW개발도구		

능 력 단 위		화면 구현	능력단위 수준	3 수준
분 류 번 호		2001020225_16v4		
능력단위 정의		화면 구현이란 UI 요구사항을 확인하여 설계한 UI 설계를 기반으로 화면을 구현하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020225_16v4.1 UI 설계 확인하기		1.1 설계된 화면과 폼의 흐름을 확인하고, 제약사항과 화면의 폼 흐름을 구현에 반영하도록 설계를 확인 할 수 있다. 1.2 UI 요구사항과 UI 표준 및 지침에 따라 설계된 메뉴 구조를 해석 할 수 있다. 1.3 구현을 위해 하위 시스템 단위의 내·외부 화면과 폼을 설계를 확인할 수 있다.	
	2001020225_16v4.2 UI 구현하기		2.1 소프트웨어 아키텍처 세부 구현 지침과 UI 표준 및 지침을 반영하여, 확인된 UI 설계를 구현할 수 있다. 2.2 확인된 화면과 폼 흐름 설계에 따라, 사용자 접근성을 고려한 화면과 폼의 흐름 제어를 구현할 수 있다. 2.3 확인된 화면과 폼 흐름 설계에 따라, 감성공학 기법을 고려하여 사용자가 접하는 화면, 폼, 메뉴, 흐름을 구현할 수 있다. 2.4 구현된 화면, 폼, 메뉴, 흐름을 테스트할 수 있는 테스트 케이스를 작성하고 단위 테스트를 수행하기 위한 테스트 조건을 명세화 할 수 있다.	
관련 지식	- 구현에 요구되는 아키텍처의 이해 - 유용성 개념 및 적용원리 - UI 기획/설계/구현 절차 - UI 표준 및 지침 - 유즈케이스에 대한 이해 - 데이터 타입 특성 - 사용자 기반 메뉴 구조 작성 방법 - 감성공학에 대한 이해 - HTML 개념과 적용원리 - CSS의 개념과 적용원리 - 자바스크립트의 개념과 적용원리 - 설계 산출물의 이해			
평가 시설· 장비	- 컴퓨터 - 문서작성도구 - SW개발도구			

능 력 단 위		애플리케이션 테스트 관리		능력단위 수준		5 수준	
분 류 번 호		2001020226_16v4					
능력단위 정의		애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력이다.					
평 가 방 법		지필평가: 복합형		시 간		60분	
		실무평가: 수행평가		시 간		120분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020226_16v4.1 애플리케이션 테스트 케이스 설계하기		1.1 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위를 결정하여 테스트케이스를 작성 할 수 있다. 1.2 개발하고자 하는 응용소프트웨어의 특성을 반영한 테스트 방식, 대상과 범위가 적용된 시나리오를 정의할 수 있다. 1.3 애플리케이션 테스트 수행에 필요한 테스트 데이터, 테스트 시작 및 종료 조건 등을 준비 할 수 있다.				
	2001020226_16v4.2 애플리케이션 개발자 통합 테스트하기		2.1 개발자 통합테스트 계획에 따라 통합 모듈 및 인터페이스가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. 2.2 개발자 통합테스트 수행 결과 발견된 결함에 대한 추이 분석을 통하여 잔존 결함을 추정할 수 있다. 2.3 개발자 통합테스트 결과에 대한 분석을 통해 테스트의 충분성 여부를 검증하고, 발견된 결함에 대한 개선 조치사항을 작성할 수 있다.				
관련 지식		<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- 해당 산업 분야에 대한 지식- 업무 특성에 대한 이해- 소프트웨어 아키텍처에 대한 이해- 국제 표준 제품 품질특성에 대한 사항- 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)에 대한 지식- 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법- 결함 관리에 대한 지식- 알고리즘- 자료구조					
평가 시설 · 장비		<ul style="list-style-type: none">- 컴퓨터- 문서작성도구- SW개발도구					

능 력 단 위		애플리케이션 테스트 수행		능력단위 수준		5 수준	
분 류 번 호		2001020227_16v4					
능력단위 정의		애플리케이션 테스트 관리란 요구사항대로 응용소프트웨어가 구현되었는지를 검증하기 위해서 테스트케이스를 작성하고 개발자 통합 테스트를 수행하여 애플리케이션의 성능을 개선하는 능력이다.					
평 가 방 법		지필평가 : 복합형		시 간		60분	
		실무평가 : 수행평가		시 간		120분	
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)				
	2001020227_16v4.1 애플리케이션 테스트 수행하기		1.1 애플리케이션 테스트 계획에 따라 서버모듈, 화면모듈, 데이터입출력, 인터페이스 등 기능단위가 요구사항을 충족하는지에 대한 테스트를 수행할 수 있다. 1.2 애플리케이션 테스트 수행으로 발견된 결함을 유형별로 기록 할 수 있다. 1.3 애플리케이션 테스트 수행 결과 발견된 결함에 대해서 원인을 분석하고 개선 방안을 도출할 수 있다.				
	2001020227_16v4.2 애플리케이션 결함 조치하기		2.1 애플리케이션 테스트 수행 결과에서 발견된 결함을 식별하고 조치에 대한 우선순위를 결정하고 적용할 수 있다. 2.2 결함이 발생한 소스를 분석하고 기존에 구현된 로직과의 연관성을 고려하여 부작용이 최소화되도록 결함을 제거할 수 있다. 2.3 애플리케이션 테스트 결과 결함 조치로 변경되는 소스의 버전을 관리하고 결함 조치 결과에 대한 이력을 관리할 수 있다.				
관련 지식	<div>- 산업 분야별 업무특성</div> <div>- 소프트웨어 아키텍처 국제 표준(IEEE 1471)</div> <div>- 국제 표준 제품 품질특성(ISO 25000)</div> <div>- 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)</div> <div>- 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법</div> <div>- 결함 관리</div> <div>- 해당 산업 분야에 대한 지식</div> <div>- 업무 특성에 대한 이해</div> <div>- 소프트웨어 아키텍처에 대한 이해</div> <div>- 국제 표준 제품 품질특성에 대한 사항</div> <div>- 테스트 지식 체계(ISTQB Syllabus, CSTE CBOK)에 대한 지식</div> <div>- 테스트 레벨(단위/통합/시스템/인수)별 테스트 접근 방법</div> <div>- 결함 관리에 대한 지식</div>						
평가 시설 · 장비	<div>- 컴퓨터</div> <div>- 문서작성도구</div> <div>- SW개발도구</div>						

능 력 단 위		소프트웨어공학 활용		능력단위 수준	3 수준
분 류 번 호		2001020228_16v4			
능력단위 정의		소프트웨어공학 활용이란 응용 소프트웨어 개발과 프로세스 적용활동의 관련된 지식을 소프트웨어의 완전성을 보장하고, 소프트웨어 품질을 평가하기 위해 CASE 도구와 형상관리를 통해 소프트웨어 공학 기술을 적용하는 능력이다.			
평 가 방 법		지필평가 : 복합형		시 간	60분
		실무평가 : 수행평가		시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)		
	2001020228_16v4.1 CASE 도구 활용하기		1.1 개발하고자 하는 응용소프트웨어에 적용할 개발방법론을 지원하는 최적의 CASE 도구를 선정할 수 있다. 1.2 CASE 도구가 제공하는 다양한 기능들 중 응용 소프트웨어 개발 시 활용할 기능을 식별할 수 있다. 1.3 CASE 도구 활용을 위한 절차와 표준을 정의하고 CASE 도구 사용 중 발생하는 이슈를 해결할 수 있다.		
	2001020228_16v4.2 애플리케이션 결함 조치하기		2.1 요구사항 명세서에 기술된 요구사항을 바탕으로 품질 표준을 정의하고 품질 평가항목과 지침을 제공할 수 있다. 2.2 요구사항 명세서에 기술된 요구사항들이 품질 표준에 따라 올바르게 기술되었는지를 검증하기 위한 품질 특성과 평가 기준을 제공할 수 있다. 2.3 개발 공정 품질 관점에서 표준 준수 여부를 확인하고, 응용소프트웨어 제품 품질 관점에서 결함을 식별하여 해결할 수 있다.		
관련 지식	<div>- 소프트웨어 개발 방법론</div> <div>- 프로젝트 관리 방법론</div> <div>- 소프트웨어 개발 표준</div> <div>- 요구공학 방법론</div> <div>- 비용산정 모델</div> <div>- 아키텍처 평가</div> <div>- 소프트웨어 개발 프레임워크에 대한 지식</div> <div>- 국제 표준 제품 품질특성에 대한 사항</div> <div>- 품질 점검 기법(리뷰, 워크스루, 인스펙션 등)</div> <div>- 품질 관리 프로세스</div> <div>- 소프트웨어 개발 표준</div> <div>- 시정 조치 관리 프로세스</div> <div>- 감사 프로세스</div>				
평가 시설· 장비	<div>- 컴퓨터</div> <div>- 문서작성도구</div> <div>- SW개발도구</div>				

능 력 단 위		소프트웨어개발 방법론 활용	능력단위 수준	6 수준
분 류 번 호		2001020229_16v4		
능력단위 정의		소프트웨어개발방법론 활용이란 응용소프트웨어 특성에 따라 정형화된 개발방법론을 선정하고, 프로젝트 특성에 맞도록 테일러링 하여 응용소프트웨어 개발에 활용하는 능력이다.		
평 가 방 법		지필평가 : 복합형	시 간	60분
		실무평가 : 수행평가	시 간	120분
평가 내용	능력단위 요소 (세 부 항 목)		수 행 준 거 (세 세 항 목)	
	2001020229_16v4.1 소프트웨어 개발방법론 선정하기		1.1 개발하여야 할 응용소프트웨어의 특성을 파악하여 정형화된 개발방법론 선정기준으로 활용할 수 있다. 1.2 정형화된 개발방법론의 특징을 고려하여 응용소프트웨어 특성에 맞는 개발방법론을 선정할 수 있다.	
	2001020229_16v4.2 소프트웨어 개발방법론 테일러링하기		2.1 프로젝트의 일정, 비용, 투입자원과 재사용 현황을 고려하여 반복전략을 개발방법론에 반영할 수 있다. 2.2 확정된 생명주기와 개발방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 및 각 단계별 종결기준을 정의할 수 있다. 2.3 응용소프트웨어 개발에 사용할 표준으로 활동 및 절차 수행에 필요한 기법과 표준, 산출물 표준 양식 및 작성 기법, 적용도구를 정립할 수 있다. 2.4 테일러링 된 적용 개발방법론에 따른 작성 산출물 유형을 확정할 수 있다.	
관련 지식	- 소프트웨어 생명주기 모델 - 소프트웨어 개발 방법론 - 요구공학 방법론 - 비용산정 모델 - 반복, 점증 프로세스 - 소프트웨어 개발 프레임워크에 대한 지식			
평가 시설· 장비	- 컴퓨터 - 문서작성도구 - SW개발도구			

