

대분류 / 20
정보통신

중분류 / 01
정보기술

소분류 / 02
정보기술개발

세분류 / 02
응용SW엔지니어링

학습모듈 / 09

09 제품소프트웨어 패키징

LM2001020209_14v2

응용SW엔지니어링 학습모듈

01. 요구사항 확인



02. 애플리케이션 설계



03. 애플리케이션 구현



04. 화면 구현



05. 데이터 입출력 구현



06. 통합 구현



07. 개발자 테스트



08. 정보시스템 이행



09. 제품소프트웨어 패키징



10. 소프트웨어공학 활용

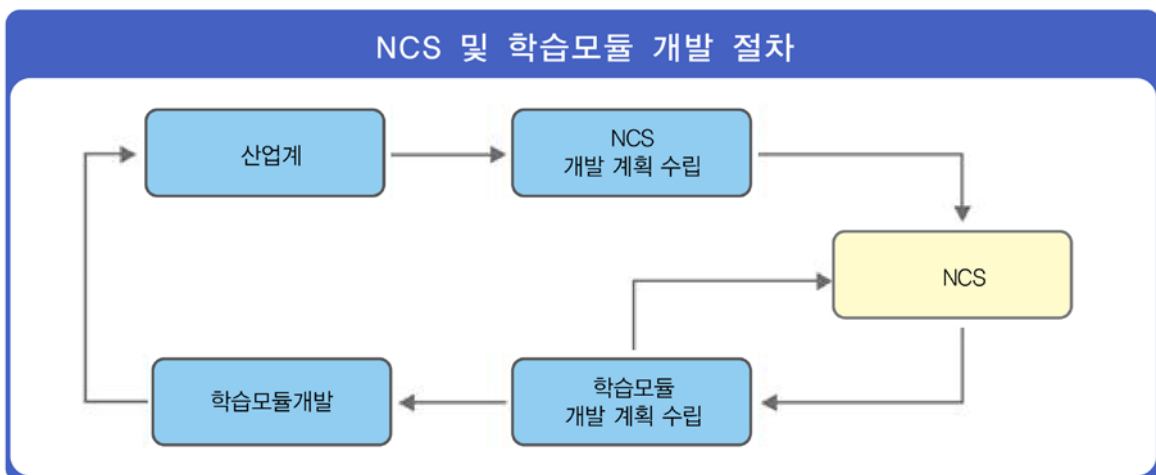


NCS 학습모듈의 이해

※ 본 학습모듈은 「NCS 국가직무능력표준」 사이트(<http://www.ncs.go.kr>) 에서 확인 및 다운로드 할 수 있습니다.

(1) NCS 학습모듈이란?

- 국가직무능력표준(NCS: National Competency Standards)이란 산업현장에서 직무를 수행하기 위해 요구되는 지식·기술·소양 등의 내용을 국가가 산업부문별·수준별로 체계화한 것으로 산업현장의 직무를 성공적으로 수행하기 위해 필요한 능력(지식, 기술, 태도)을 국가적 차원에서 표준화한 것을 의미합니다.
- 국가직무능력표준(이하 NCS)이 현장의 ‘직무 요구서’라고 한다면, NCS 학습모듈은 NCS의 능력단위를 교육훈련에서 학습할 수 있도록 구성한 ‘교수·학습 자료’입니다. NCS 학습모듈은 구체적 직무를 학습할 수 있도록 이론 및 실습과 관련된 내용을 상세하게 제시하고 있습니다.

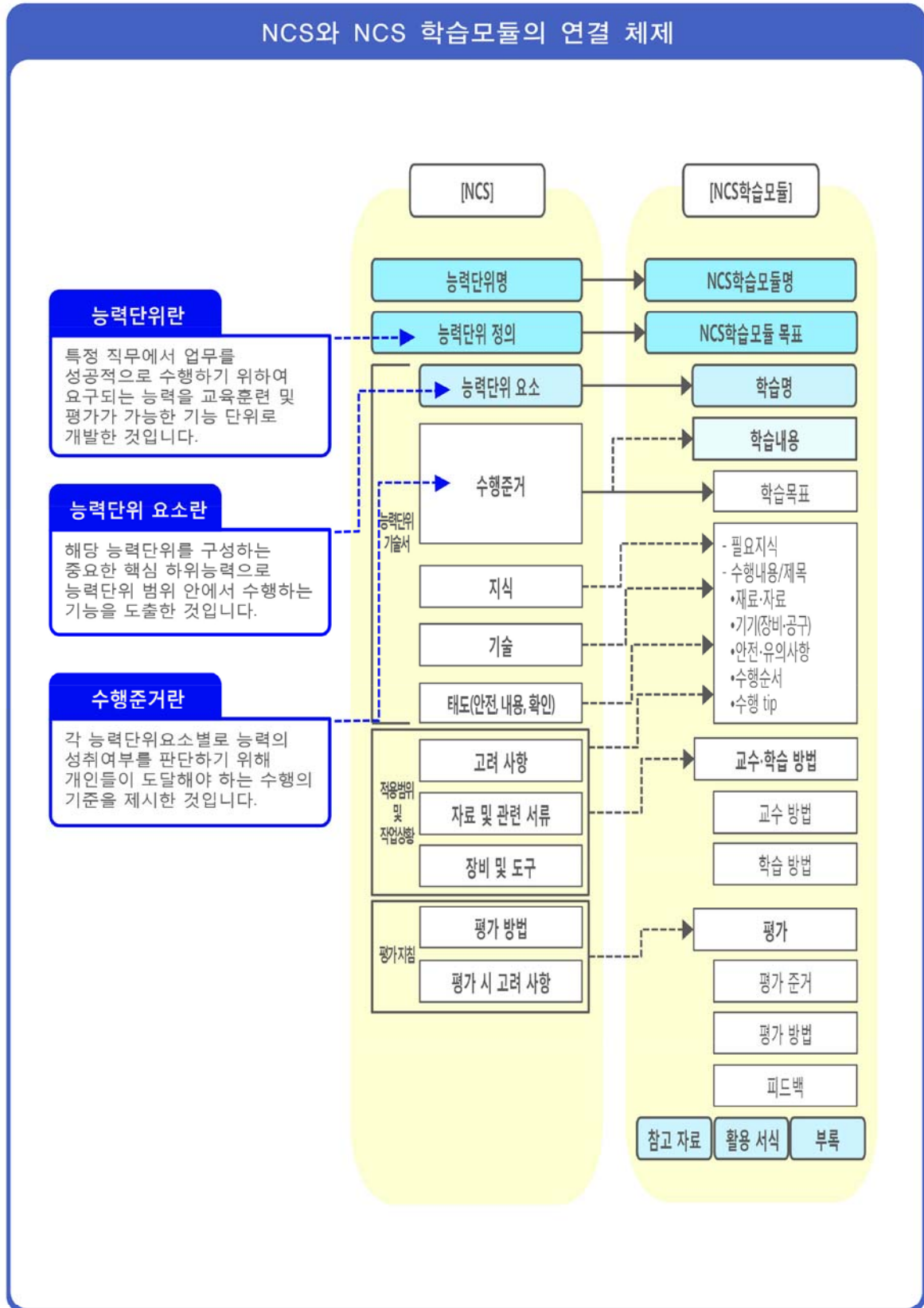


- NCS 학습모듈은 다음과 같은 특징을 가지고 있습니다.

첫째, NCS 학습모듈은 산업계에서 요구하는 직무능력을 교육훈련 현장에 활용할 수 있도록 성취목표와 학습의 방향을 명확히 제시하는 가이드라인의 역할을 합니다.

둘째, NCS 학습모듈은 특성화고, 마이스터고, 전문대학, 4년제 대학교의 교육기관 및 훈련기관, 직장교육기관 등에서 표준교재로 활용할 수 있으며 교육과정 개편 시에도 유용하게 참고할 수 있습니다.

- NCS와 NCS 학습모듈 간의 연결 체제를 살펴보면 아래 그림과 같습니다.



(2) NCS 학습모듈의 체계

- NCS 학습모듈은 1.학습모듈의 위치, 2.학습모듈의 개요, 3.학습모듈의 내용 체계, 4.참고 자료, 5.활용 서식/부록 으로 구성되어 있습니다.

1. NCS 학습모듈의 위치

- NCS 학습모듈의 위치는 NCS 분류 체계에서 해당 학습모듈이 어디에 위치하는지를 한 눈에 볼 수 있도록 그림으로 제시한 것입니다.

예시 : 이·미용 서비스 분야 중 네일미용 세분류

NCS-학습모듈의 위치

대분류	이용·숙박·여행·오락·스포츠
중분류	이·미용
소분류	아·미용 서비스

세분류	능력단위	학습모듈명
헤어미용	네일 샵 위생 서비스	네일샵 위생서비스
피부미용	네일 화장을 제거	네일 화장을 제거
메이크업	네일 기본 관리	네일 기본관리
네일미용	네일 랩	네일 랩
이용	네일 팁	네일 팁
	젤 네일	젤 네일
	아크릴릭 네일	아크릴 네일
	평면 네일아트	평면 네일아트
	융합 네일아트	융합 네일아트
	네일 샵 운영관리	네일샵 운영관리

학습모듈은

NCS 능력단위 1개당 1개의 학습모듈 개발을 원칙으로 합니다. 그러나 필요에 따라 고용 단위 및 교과단위를 고려하여 능력단위 몇 개를 묶어서 1개의 학습모듈로 개발할 수 있으며, NCS 능력단위 1개를 여러 개의 학습모듈로 나누어 개발할 수도 있습니다.

2. NCS 학습모듈의 개요

구 성

- NCS 학습모듈 개요는 학습모듈이 포함하고 있는 내용을 개략적으로 설명한 것으로서 **학습모듈의 목표**, **선수 학습**, **학습모듈의 내용 체계**, **핵심 용어**로 구성되어 있습니다.

학습모듈의 목표	해당 NCS 능력단위의 정의를 토대로 학습목표를 작성한 것입니다.
선수 학습	해당 학습모듈에 대한 효과적인 교수·학습을 위하여 사전에 이수해야 하는 학습모듈, 학습 내용, 관련 교과목 등을 기술한 것입니다.
학습모듈의 내용 체계	해당 NCS 능력단위요소가 학습모듈에서 구조화된 방식을 제시한 것입니다.
핵심 용어	해당 학습모듈의 학습 내용, 수행 내용, 설비·기자재 등 가운데 핵심적인 용어를 제시한 것입니다.

활 용 안 내

예시 : 네일미용 세분류의 ‘네일 기본관리’ 학습모듈

네일 기본관리 학습모듈의 개요

학습모듈의 목표

고객의 네일 보호와 미적 요구 충족을 위하여 효과적인 네일 관리로 프리에지 형태 만들기, 큐티클 정리하기, 컬러링하기, 보습제 도포하기, 마무리를 할 수 있다.

선수학습

네일숍 위생서비스(JM1201010401_14v2)

학습모듈의 내용체계

학습	학습내용	NCS 능력단위요소		
		코드번호	요소명칭	수준
1. 프리에지 형태 만들기	1-1. 네일 파일에 대한 이해와 활용 1-2. 프리에지 형태 파일링	1201010403_12v2.1	프리에지 모양 만들기	3
2. 큐티클 정리하기	2-1. 네일 기본관리 매뉴얼 이해 2-2. 큐티클 관리	1201010403_14v2.2	큐티클 정리하기	3
3. 컬러링하기	3-1. 컬러링 매뉴얼 이해 3-2. 컬러링 방법 선정과 작업 3-3. 쉘 컬러링 작업	1201010403_14v2.3	컬러링	3
4. 보습제 도포하기	4-1. 보습제 선정과 도포 4-2. 각질제거	1201010403_14v2.4	보습제 바르기	2
5. 네일 기본관리 마무리하기	5-1. 유폴기 제거 5-2. 네일 기본관리 마무리와 정리	1201010403_14v2.5	마무리하기	3

핵심 용어

프리에지, 니퍼, 푸서, 플리시, 네일 파일, 스웨이형, 스웨이 오프형, 라운드형, 오발형, 포인트형

학습모듈의 목표는

학습자가 해당 학습모듈을 통해 성취해야 할 목표를 제시한 것으로, 교수자는 학습자가 학습모듈의 전체적인 내용흐름을 파악할 수 있도록 지도하는 것이 필요합니다.

선수학습은

교수자나 학습자가 해당 모듈을 교수 또는 학습하기 이전에 이수해야 할 학습내용, 교과목, 핵심 단어 등을 표기한 것입니다. 따라서 교수자는 학습자가 개별 학습, 자기 주도 학습, 방과 후 활동 등 다양한 방법을 통해 이수할 수 있도록 지도하는 것이 필요합니다.

핵심 용어는

학습모듈을 통해 학습되고 평가되어야 할 주요 용어입니다. 또한 당해 모듈 또는 타 모듈에서도 핵심 용어를 사용하여 학습내용을 구성할 수 있으며, 「NCS 국가 직무능력표준」 사이트(www.ncs.go.kr)에서 색인(찾아보기) 중 하나로 이용할 수 있습니다.

3. NCS 학습모듈의 내용 체계

구 성

- NCS 학습모듈의 내용은 크게 **학습**, **학습 내용**, **교수·학습 방법**, **평가** 로 구성되어 있습니다.

학습	해당 NCS 능력단위요소 명칭을 사용하여 제시한 것입니다. 학습은 크게 학습 내용, 교수·학습 방법, 평가로 구성되며 해당 NCS 능력단위의 능력단위 요소별 지식, 기술, 태도 등을 토대로 학습 내용을 제시한 것입니다.
학습 내용	학습 내용은 학습 목표, 필요 지식, 수행 내용으로 구성하였으며, 수행 내용은 재료·자료, 기기(장비·공구), 안전·유의 사항, 수행 순서, 수행 tip으로 구성한 것입니다. 학습모듈의 학습 내용은 업무의 표준화된 프로세스에 기반을 두고 실제 산업현장에서 이루어지는 업무활동을 다양한 방식으로 반영한 것입니다.
교수·학습 방법	학습 목표를 성취하기 위한 교수자와 학습자 간, 학습자와 학습자 간의 상호 작용이 활발하게 일어날 수 있도록 교수자의 활동 및 교수 전략, 학습자의 활동을 제시한 것입니다.
평가	평가는 해당 학습모듈의 학습 정도를 확인할 수 있는 평가 준거, 평가 방법, 평가 결과의 피드백 방법을 제시한 것입니다.

활 용 안 내

예시 : 네일미용 세분류의 ‘네일 기본관리’ 학습모듈의 내용

학습 1	프리에지 형태 만들기(LM1201010403_14v2.1)
학습 2	큐티클 정리하기(LM1201010403_14v2.2)
학습 3	컬러링하기(LM1201010403_14v2.3)
학습 4	보습제 도포하기(LM1201010403_14v2.4)
학습 5	네일 기본관리 마무리하기(LM1201010403_14v2.5)

학습은

해당 NCS 능력단위요소 명칭을 사용하여 제시하였습니다. 학습은 일반교과의 '대단원'에 해당되며, 모듈을 구성하는 가장 큰 단위가 됩니다. 또한 완성된 직무를 수행하기 위한 가장 기본적인 단위로 사용할 수 있습니다.

학습내용은

요소 별 수행준거를 기준으로 제시하였습니다. 일반교과의 '중단원'에 해당합니다.

학습목표는

모듈 내의 학습내용을 이수했을 때 학습자가 보여줄 수 있는 행동수준을 의미합니다. 따라서 일반 수업시간의 과목목표로 활용할 수 있습니다.

필요지식은

해당 NCS의 지식을 토대로 해당 학습에 대한 이해와 성과를 높이기 위해 알아야 할 주요 지식을 제시하였습니다. 필요지식은 수행에 꼭 필요한 핵심 내용을 위주로 제시하여 교수자의 역할이 매우 중요하며, 이후 수행순서 내용과 연계하여 교수·학습으로 진행할 수 있습니다.

3-1. 컬러링 매뉴얼 이해

학습목표

- 고객의 요구에 따라 네일 폴리시 색상의 침착을 막기 위한 베이스코트를 아주 얇게 도포할 수 있다.
- 작업 매뉴얼에 따라 네일 폴리시를 얼룩 없이 균일하게 도포할 수 있다.
- 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다.

필요 지식 /

□ 컬러링 매뉴얼

컬러링 작업 전, 이세톤 또는 네일 폴리시 리무버를 사용하여 손톱표면과 큐티클 주변, 손톱 밑 부분까지 깨끗하게 유분기를 제거해야 한다. 컬러링의 순서는 Base coating 1회 → Polishing 2회 → 컬러수정 → Top coating 1회 → 최종수정의 순서로 한다. 베이스코트는 착색을 방지하고 발림성 향상을 위해 가장 먼저 도포하며 컬러링의 마지막에 컬러의 유지와 광택을 위해 톱코트를 도포한다. 네일 보강제(Nail Strengthner)를 바를 시에는 베이스코트를 도포하기 전에 사용한다.

수행 내용 / 컬러링 매뉴얼 실습하기

재료·자료

- 컬러링 관련 네일 미용 자료들
- 컬러바구니, 베이스코트, 네일 폴리시, 톱코트, 오렌지우드스티, 탈지면, 폴리시리무버, 디스펜서 등

기기(장비·공구)

- 컴퓨터, 빔 프로젝터, 스크린 등

안전·유의사항

- 컬러링 재료들의 냄새를 직접적으로 맡지 않도록 유의한다.
- 컬러링 제품들이 대부분 유리병에 들어 있기 때문에 깨지지 않도록 각별히 조심한다.
- 컬러링 제품들은 상온에 마르기 때문에 개봉 후 뚜껑을 잘 닫도록 한다.

수행 순서

[1] 네일 폴리시를 바르게 잡는다.

1. 손바닥에 네일 폴리시를 놓고 약지 소지를 이용하여 네일 폴리시를 잡는다.
2. 폴리시를 왼 손의 엄지와 검지와 고객의 작업손가락을 잡는다.
3. 폴리시를 왼 손의 중지 손가락을 굳게 펴서 받침대가 되도록 한다.
4. 반대편 손으로 네일 폴리시의 뚜껑을 열고 소지 손가락을 펴서 네일 폴리시를 왼 중지 손가락 위에 받쳐놓는다.
5. 다양한 형태의 폴리시를 잡아본다.

수행 tip

- 흰색이 많이 섞인 네일 폴리시의 경우는 붓의 각도를 높이 세워서 빠르게 브러시 작업을 해야 붓 자국이 나지 않는다.
- 컬러링은 기본 2회 정도이나 컬러에 따른 도포량과 컬러감에 따라 1~3회 사이로 증감할 수 있다.

수행 내용은

모듈에 제시한 것 중 기술(Skill)을 습득하기 위한 실습 과제로 활용할 수 있습니다.

재료·자료는

수행 내용을 수행하는데 필요한 재료 및 준비물로 실습 시 필요 준비물로 활용할 수 있습니다.

기기(장비·공구)는

수행 내용을 수행하는데 필요한 기본적인 장비 및 도구를 제시하였습니다. 제시된 기기 외에도 수행에 필요한 다양한 도구나 장비를 활용할 수 있습니다.

안전·유의사항은

수행 내용을 수행하는데 안전상 주의해야 할 점 및 유의사항을 제시하였습니다. 수행 시 유념해야 하며, NCS의 고려사항도 추가적으로 활용할 수 있습니다.

수행 순서는

실습과제의 진행 순서로 활용할 수 있습니다.

수행 tip은

수행 내용에서 수행의 수월성을 높일 수 있는 아이디어를 제시하였습니다. 따라서 수행tip은 지도상의 안전 및 유의사항 외에 전반적으로 적용되는 주안점 및 수행과제 목적에 대한 보충설명, 추가사항 등으로 활용할 수 있습니다.

학습3 교수·학습 방법

교수·학습 방법은

학습목표를 성취하는데 필요한 교수 방법과 학습 방법을 제시하였습니다.

교수 방법

- 컬러링 제품의 성분과 컬러별 정도의 차이, 베이스코트와 톱코트의 역할, 폴리시 잡는 방법, 큐어링 시간 등의 내용을 화면 자료와 함께 설명한다.
- 서식지를 활용하여 네일 컬러링 방법을 그림으로 그려 보게 한 뒤, 다양한 컬러링의 매뉴얼을 그려서 숙지하도록 한다.
- 겔 컬러링 시 주의사항을 계속 숙지시키도록 하며, 큐어링 시간에 대해 작성하도록 한다.

교수 방법은

해당 학습활동에 필요한 학습내용, 학습내용과 관련된 학습자료명, 자료 형태, 수행내용의 진행 방식 등에 대하여 제시하였습니다. 또한 학습자의 수업참여도를 제고하기 위한 방법 및 수업진행상 유의사항 등도 제시하였습니다. 선수학습이 필요한 학습을 학습자가 숙지하였는지 교수자가 확인하는 과정으로 활용할 수도 있습니다.

학습 방법

- 컬러링을 위한 재료의 필요성과 사용방법을 숙지하고 컬러링 매뉴얼 과정에 맞추어 작업 내용을 이해한다.
- 컬러링의 다양성에 대한 용어를 숙지하고 진행과정에 맞추어 내용을 작업한다.
- 겔 컬러링 시 적합한 큐어링 시간을 선택해서 큐어링 해본다.

학습 방법은

해당 학습활동에 필요한 학습자의 자기주도적 학습 방법을 제시하였습니다. 또한 학습자가 숙달해야 할 실기능력과 학습과정에서 주의해야 할 사항 등으로 제시하였습니다. 학습자가 학습을 이수하기 전에 반드시 숙지해야 할 기본 지식을 학습하였는지 스스로 확인하는 과정으로 활용할 수 있습니다.

학습3 평가

평가 준거

- 평가자는 학습자가 학습 목표 및 평가 항목에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가해야 한다.

학습내용	평가항목	성취수준		
		상	중	하
컬러링 매뉴얼 이해	<ul style="list-style-type: none"> 고객의 요구에 따라 네일 폴리시 색상의 질감을 만들기 위한 베이스코트를 아주 얇게 도포할 수 있다. 작업 매뉴얼에 따라 네일 폴리시를 일찍 얇이 균일하게 도포할 수 있다. 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다. 			

평가 방법

- 작업장 평가

학습내용	평가항목	성취수준		
		상	중	하
컬러링 매뉴얼 이해	<ul style="list-style-type: none"> 고객의 요구에 따라 네일 폴리시 색상의 질감을 만들기 위한 베이스코트를 아주 얇게 도포할 수 있다. 작업 매뉴얼에 따라 네일 폴리시를 일찍 얇이 균일하게 도포할 수 있다. 작업 매뉴얼에 따라 네일 폴리시 도포 후 컬러 보호와 광택 부여를 위한 톱코트를 바를 수 있다. 			

피드백

- 작업장 평가
 - 작업 결과물을 확인하여 수정사항을 제시하고 수정 부분을 인지하도록 한다.

평가는

해당 NCS 능력단위 평가방법과 평가 시 고려 사항을 준용하여 작성하였습니다. 교수자 및 학습자가 평가항목 별 성취수준을 확인하는데 활용할 수 있습니다.

평가 준거는

학습자가 해당 학습을 어느 정도 성취하였는지를 평가하기 위한 기준을 제시하고 있습니다. 학습목표와 연계하여 단위수업 시간에 평가항목 별 성취수준을 평가하는데 활용할 수 있습니다.

평가 방법은

NCS 능력단위의 평가방법을 준용하였으며, 평가 준거에 따른 평가방법을 2개 이상 제시하였습니다. 평가방법으로는 포트폴리오, 문제해결 시나리오, 서술형 시험, 논술형 시험, 사례연구, 평가자 체크리스트, 작업장 평가 등이 있으며, NCS의 능력단위 요소 별 수행 수준을 평가하는데 가장 적절한 방법을 선정하여 활용할 수 있습니다.

피드백은

평가 후에 학습자들에게 평가 결과를 피드백하여 부족한 부분을 알려주고, 학습 결과가 미진한 경우, 해당 부분을 다시 학습하여 학습목표를 달성하는 데 활용할 수 있습니다.

4. 참고 자료

참고자료

- 김미원(2011). 『Nail Study』. 서울: 사)한국네일저서서비스협회.
- 민광경(2015). 『미용사(네일)평가』. 서울: 예문사.
- 박은국(2014). 『네일미용』. 서울: 정담미디어.

참고자료는


해당 학습모듈의 필요지식에 대한 출처와 인용한 참고자료 및 사이트를 제시하였습니다.

5. 활용 서식/부록

활용서식

프리에지 형태 실습지

1. 프리에지 형태의 이해

모양	이름	특징
	{ Square nail }	-강한 느낌의 사각형태 -네일의 양끝 모서리 부분이 90° 사각의 형태이다. { } -발톱의 형태 활용 -새인성 발톱의 보정시에 적용

활용서식은

평가 서식, 실습시트 등 교수학습 시 활용 가능한 다양한 서식들로 구성하였습니다. 과제 진행에서 평가에 이르기까지 필요한 서식을 해당 학습모듈의 특성에 맞춰 개발하거나 기존의 양식을 활용하여 제시하였습니다.

부록

네일 기본관리 도구와 재료 목록

목록	비고	준비
위생가운	흰색	작업자 착용
위생 마스크	흰색	작업자 착용
보호안경	투명한 렌즈 (안경으로 대체 가능)	작업자 착용
세면접리함	재질, 색상 무관	작업대

부록은

활용서식 이외에 교수학습과정에서 참고할 수 있는 자료가 있는 경우 제시하였습니다.

[NCS-학습모듈의 위치]

대분류	정보 통신	
중분류	정보 기술	
소분류	정보 기술 개발	

세분류		
SW아키텍처	능력단위	학습모듈명
응용SW 엔지니어링	요구 사항 확인	요구 사항 확인
시스템 엔지니어링	애플리케이션 설계	애플리케이션 설계
DB엔지니어링	애플리케이션 구현	애플리케이션 구현
NW엔지니어링	화면 구현	화면 구현
보안엔지니어링	데이터 입출력 구현	데이터 입출력 구현
UI/UX 엔지니어링	통합 구현	통합 구현
시스템SW 엔지니어링	개발자 테스트	개발자 테스트
	정보시스템 이행	정보 시스템 이행
	제품소프트웨어 패키징	제품소프트웨어 패키징
	소프트웨어공학 활용	소프트웨어공학 활용

차 례

학습모듈의 개요	1
학습 1. 제품 소프트웨어 패키징하기	
1-1. 사용자 중심의 패키징 수행	3
1-2. 버전을 고려한 제품 릴리즈 노트 작성	9
1-3. 패키징 도구를 활용한 설치, 배포 수행	14
• 교수·학습 방법	20
• 평가	21
학습 2. 제품 소프트웨어 매뉴얼 작성하기	
2-1. 제품 소프트웨어 설치 매뉴얼 구성 수립 및 작성	23
2-2. 제품 소프트웨어 사용자 매뉴얼 구성 수립 및 작성	31
2-3. 제품 소프트웨어 매뉴얼의 배포용 미디어 제작	40
• 교수·학습 방법	45
• 평가	47
학습 3. 제품 소프트웨어 버전 관리하기	
3-1. 제품 소프트웨어 버전 등록	49
3-2. 제품 소프트웨어 버전 관리 도구 사용	55
3-3. 제품 소프트웨어 버전 관리 도구 활용	63
• 교수·학습 방법	68
• 평가	70
참고 자료	72
활용 서식	73

제품소프트웨어 패키징 학습מוד의 개요

학습מוד의 목표

개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품 소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전 관리를 수행할 수 있다.

선수학습

공통 모듈 설계, 내외부 연계 모듈 구현, 개발 환경 구축, 정보 시스템 이행(2001020208_14v2)

학습מוד의 내용 체계

학습	학습 내용	NCS 능력단위 요소		
		코드 번호	요소 명칭	수준
1. 제품 소프트웨어 패키징하기	1-1 사용자 중심의 패키징 수행 1-2 버전을 고려한 제품 릴리즈 노트 작성 1-3 패키징 도구를 활용한 설치, 배포 수행	2001020209_14v2.1	제품 소프트웨어 패키징하기	3
2. 제품 소프트웨어 매뉴얼 작성하기	2-1 제품 소프트웨어 설치 매뉴얼 구성 수립 및 작성 2-2 제품 소프트웨어 사용자 매뉴얼 구성 수립 및 작성 2-3 제품 소프트웨어 매뉴얼의 배포용 미디어 제작	2001020209_14v2.2	제품 소프트웨어 매뉴얼 작성하기	3
3. 제품소프트웨어 버전 관리하기	3-1 제품 소프트웨어 버전 등록 3-2 제품 소프트웨어 버전 관리 도구 사용 3-3 제품 소프트웨어 버전 관리 도구 활용	2001020209_14v2.3	제품 소프트웨어 버전 관리하기	2

핵심 용어

소프트웨어 패키징, 소프트웨어 빌드, 모듈화, 형상 관리, 버전 관리, 설치 매뉴얼, 사용자 매뉴얼, DRM, BACKUP, D/R

학습 1

제품 소프트웨어 패키징하기 (LM2001020209_14v2.1)

학습 2

제품 소프트웨어 매뉴얼 작성하기(LM2001020209_14v2.2)

학습 3

제품 소프트웨어 버전 관리하기(LM2001020209_14v2.3)

1-1. 사용자 중심의 패키징 수행

학습 목표

- 신규 개발, 변경, 개선된 제품 소프트웨어의 소스들로부터 모듈들을 빌드하고 고객의 편의성을 고려하여 패키징할 수 있다.

필요 지식 /

① 제품 소프트웨어 패키징의 개요

제품 소프트웨어 패키징은 고객 편의성 중심으로 진행되며, 이를 위한 매뉴얼 및 버전 관리를 포함한다. 따라서 전체적인 개념을 먼저 이해하고 작업을 진행한다.

1. 제품 소프트웨어 패키징의 개념

제품 소프트웨어 패키징은 개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품 소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전 관리를 수행하는 능력이다.

2. 제품 소프트웨어 적용상의 특성

- (1) 제품 소프트웨어는 개발자가 아닌 사용자 중심으로 진행된다.
- (2) 신규 및 변경 개발 소스를 식별하고, 이를 모듈화하여 상용 제품으로 패키징한다.
- (3) 고객의 편의성을 위해, 신규/변경 이력을 확인하고, 이를 버전 관리 및 릴리즈 노트를 통해 지속적으로 관리해 간다.
- (4) 사용자의 실행 환경을 이해하고, 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 분류하여 패키징이 진행된다.

② 제품 소프트웨어 패키징을 위한 모듈 빌드

제품 소프트웨어 패키징은 모듈들을 통해 수행되므로, 최초 소스 개발 시 해당 모듈의 개념 및 모듈화를 통해 제품 소프트웨어 패키징을 가져간다. 소프트웨어의 모듈 및 모듈화를 이해하고 이를 빌드하는 기법을 통해 패키징을 성공적으로 수행한다.

1. 소프트웨어의 모듈 및 패키징

(1) 소프트웨어 모듈의 이해

(가) 모듈 및 모듈화

- 소프트웨어 설계에서 기능단위로 분해하고 추상화 되어 재사용 및 공유 가능한 수준으로 만들어진 단위를 모듈로 규정하고, 소프트웨어의 성능을 향상시키거나 시스템의 디버깅, 시험, 통합 및 수정을 용이하도록 하는 소프트웨어 설계 기법을 모듈화라고 한다.

(나) 모듈화의 장점

- 프로그램의 효율적인 관리 및 성능 향상
- 전체적인 소프트웨어 이해의 용이성 증대 및 복잡성 감소
- 소프트웨어 시험, 통합, 수정 시 용이성 제공
- 기능의 분리가 가능하고 인터페이스가 단순
- 오류의 파급 효과를 최소화
- 모듈의 재사용 가능으로 개발과 유지보수가 용이

(다) 모듈화의 목표

- 모듈 간 결합도의 최소화(Loose Coupling)
- 모듈 내 요소들간의 응집도 최대화(Strong Cohesion)

(2) 제품 소프트웨어의 모듈 및 패키징

제품 소프트웨어에서는 모듈의 개념을 정확하게 이입하고, 이에 맞는 기능 단위로 패키징하는 것이 필요하다. 패키징 배포 시 제품 소프트웨어의 성능을 향상시킬 수도 있고 배포 전 시험 및 수정 등의 작업 진행에서도 모듈 단위로 모든 것을 분류하여 작업을 진행한다.

2. 제품 소프트웨어 모듈 빌드 기법

(1) 소프트웨어 빌드

소프트웨어 빌드(Software Build)는 소스 코드 파일을 컴퓨터에서 실행할 수 있는 제

품 소프트웨어의 단위로 변환하는 과정을 말하거나 그에 대한 결과물을 일컫는다. 소프트웨어 빌드는 소스 코드 파일이 실행 코드로 변환되는 컴파일 과정을 핵심으로 수행되며, 빌드에 따른 결과물에 대한 상세 확인이 필요하다.

(2) 제품 소프트웨어를 위한 빌드 기법

소프트웨어 빌드 시스템의 기본 개념을 알고, 빌드의 실행 단위 컴파일, 이를 위한 빌드 도구의 특징 및 사례들의 사전 확인이 필요하다. 빌드 도구는 이를 도와주는 유용한 유틸리티이며 이를 활용하여 컴파일 이외에도 제품 소프트웨어 완성을 위해 다양한 일을 할 수 있다. 대표적인 도구들로는 Ant, Make, Maven, Gradle 등의 빌드 도구들이 있다. 또 빌드 과정에 대한 문서화를 체크해 두면 다양한 이력에 대한 관리가 가능하다.

③ 사용자 중심의 패키징 작업 이해

제품 소프트웨어 패키징은 사용자를 위해 진행되는 작업으로서, 고객 편의성 및 사용자 실행 환경을 우선 고려하여 진행한다.

1. 사용자 실행 환경의 이해

고객 편의성을 위해 사용자 실행 환경을 우선 고려하여 패키징을 진행한다. 여기서의 사용자 환경은 OS부터 시작하여 실행 환경, 시스템 사양 및 고객의 사용 방법까지 상세 분류하여 실행 환경을 사전 정의한다. 만약 여러 가지 실행 환경이 나오게 된다면 해당 경우에 맞는 배포본을 분류하여 패키징 작업을 여러 번 수행할 수도 있다.

2. 사용자 관점에서의 패키징 고려 사항

고객 편의성을 위해 사용자 환경을 확인하여 다음과 같이 여러 가지 케이스를 사전에 고려한다.

- (1) 사용자의 시스템 환경 즉, OS CPU 메모리 등의 수행 최소 환경을 정의한다.
- (2) 사용자가 직관적으로 확인할 수 있는 UI(User Interface)를 제공하고, 매뉴얼과 일치시켜 패키징 작업한다.
- (3) 제품 소프트웨어는 하드웨어와 함께 통합 적용될 수 있도록, 패키징은 Managed Service 형태로 제공되는 것이 좋다.
- (4) 제품 소프트웨어는 고객 편의성을 위해 안정적 배포가 가장 중요하고, 다양한 사용자 군(群)의 요구 사항을 반영하기 위해 항상 패키징의 변경 및 개선 관리를 고려하여 패키징 배포한다.

<핵심용어 정의>

제품 소프트웨어 패키징
개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품 소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전 관리를 수행하는 능력
제품 소프트웨어 매뉴얼
제품 소프트웨어 개발 단계부터의 적용 기준이나 패키징 이후 설치 및 사용자 측면의 주요 내용 등을 문서로 기록한 것이며, 사용자 중심의 기능 및 방법을 나타낸 설명서와 안내서
제품 소프트웨어 버전 관리
기본적으로 작성된 패키지의 변경 내용을 관리하고, SW의 변화를 시간에 따라 기록하며 특정 시점의 버전을 다시 꺼내올 수 있도록 관리하는 체계

수행 내용 / 사용자 중심의 모듈 패키징 작업 수행하기

재료 · 자료

- 신규 개발 소스
- 모듈 정의서, UI 정의서
- 사용자 환경과 맞는 최소 시스템 사양

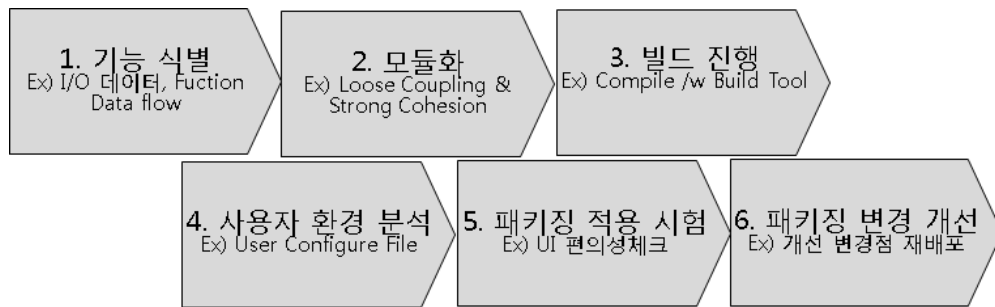
기기(장비 · 공구)

- 컴퓨터, 인터넷
- 모듈 빌드 프로그램

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서



[그림 1-1] 패키징 작업 수행

① 신규 개발 소스의 목적 및 기능을 식별한다.

1. 기능 수행을 위한 입출력 데이터 : 소스 기능을 통해 처리되는 int, char형 데이터이다.
2. 전체적인 기능 정의 및 데이터 흐름 : 소스 내부 메인 함수의 기능 식별, 관련 데이터의 흐름 및 출력 절차를 확인한다.
3. Function 단위 및 Output에 대한 상세 정의 : 메인 함수 이외의 호출 함수를 정의하고 이에 대한 출력값을 식별한다.

② 모듈 단위 분류 및 모듈화를 순서에 맞게 진행한다.

1. 수행을 위한 기능 단위 및 서비스 분류 : 모듈로 분류할 수 있는 기능 및 서비스 단위로 분류한다.
2. 패키징 개념에서의 기능의 공유와 재활용 분류 : 여러 번 호출되어 지는 공유, 재활용 함수를 분류한다.
3. 모듈 간 결합도와 응집도 (Loose Coupling & Strong Cohesion) 식별 : 모듈화를 위해 결합도와 응집도를 식별해내고 모듈화 수행을 준비한다.

③ 제품 소프트웨어의 빌드 도구를 활용한 빌드를 단위별 진행한다.

1. 신규 개발 소스 및 컴파일 결과물 준비 : 빌드 진행을 위한 신규 소스 및 결과물을 준비한다.
2. 정상 기능 단위 및 서비스 분류 : 정상으로 빌드가 되는 기능 및 서비스를 사전에 분류한다.
3. 빌드 도구 확인 및 정상 수행 : 빌드 도구의 사전 선택 확인 및 빌드 도구를 통한 빌드를 수행한다.

4. 컴파일 이외의 도구의 다양한 기능 확인 : 컴파일 이외의 에디터, CI 도구 등 관련 도구들에 대한 추가 학습을 진행한다.

④ 고객 편의성을 위한 사용자의 요구 사항 및 사용 환경을 사전에 분석한다.

1. 최소 사용자 환경 사전 정의 : 패키징 수행 시 실제 사용할 User의 사용자 환경을 사전에 정의한다.
2. 모듈 단위의 여러 가지 기능별 사용자 환경 테스트 : 수행한 모듈화의 결과에 따른 여러 가지 사용자 테스트를 수행한다.

⑤ 최종 패키징에 대해서 사용자 입장에서의 불편한 점을 체크한다.

1. 사용자 환경에서의 패키징 적용 시험 : 사용자 환경과 똑같은 상황으로 패키징을 적용하여 SW테스팅을 수행한다.
2. UI 및 시스템 상의 편의성 체크 : SW가 UI 및 시스템 환경과 맞지 않는 것이 있는지, 불편한 부분이 있는지 사전에 체크한다.

⑥ 사용자 입장을 반영하여 패키징에서 변경 및 개선을 진행한다.

1. 패키징 적용 시 변경점 도출

사용자 입장을 반영하여 재패키징을 대비해 변경 부분을 정리한다.

2. 최소 사용자 환경에서 서비스 가능한 수준의 개선

현재 사용자 환경 내에서 가능한 최소 수준의 개선 포인트를 도출한다.

3. 모듈 및 빌드 수정을 통한 개선 버전의 재배포를 실습

도출된 변경점을 기준으로 모듈, 빌드 수정을 하고 재배포를 수행한다.

수행 tip

- 패키징 작업 시 최소 신규 소스에 대한 분석 및 이에 대한 모듈화, 빌드의 작업을 흐름에 맞게 순차적으로 진행할 수 있는지가 중요하다. 가장 기본이 되어야 할 고객 편의성 측면과 이에 따른 개선 사항을 도출해 내는 것도 중요한 포인트이다.

1-2. 버전을 고려한 제품 릴리즈 노트 작성

학습 목표

- 이전 릴리즈 이후의 변경, 개선 사항을 포함하여 신규 패키징한 제품 소프트웨어에 대한 릴리즈 노트를 작성할 수 있다.

필요 지식 /

① 패키징에서의 릴리즈 노트의 파악

제품 소프트웨어 패키징에서의 릴리즈 노트는, 소프트웨어 제품이 개발 또는 테스트 상태이거나, 제품 베타 버전 등의 상태일 경우, 업데이트가 해제될 때 등 다양한 케이스에 대해서 실무에서 작성되어 왔다. 제품 소프트웨어 패키징의 경우, 앞 단위에서 학습한 것처럼 고객 편의성을 고려하여 릴리즈 노트를 작성하고 고객에게 전달된다.

1. 릴리즈 노트의 개념

릴리즈 노트는 조직의 최종 사용자인 고객과 잘 정리된 릴리즈 정보를 공유하는 문서이다. 상세 서비스를 포함하여 회사가 제공하는 제품을 만들어 수정, 변경 또는 개선하는 일련의 작업들이며, 릴리즈 정보들이 이러한 문서를 통해 제공된다. 이 정보들은 철저하게 테스트를 진행하고, 개발 팀에서 제공하는 사양에 대해 최종 승인된 후 문서를 통해 배포된다.

2. 릴리즈 노트의 중요성

(1) 릴리즈 노트에는 테스트 결과와 정보가 포함된다.

사용자에게 최종 배포된 릴리즈 노트를 보면 테스트가 어떻게 진행됐는지, 개발 팀의 제공 사양을 얼마나 준수했는지를 확인해 볼 수 있다.

(2) 사용자에게 보다 더 확실한 정보를 제공한다.

(3) 기본적으로 전체적인 제품의 수행 기능 및 서비스의 변화를 공유한다.

(4) 자동화 개념과 함께하여 적용할 수 있다. 이를 통해 전체적인 버전 관리 및 릴리즈 정보를 체계적으로 관리할 수 있다.

② 릴리즈 노트 작성 시 고려 사항

릴리즈 노트는 개발 팀에서 Ownership을 가지고 직접 작성하여야 한다. 현재 시제로 작성되어야 하며, 명확하고 정확하며 완전한 정보를 제공한다.

그리고 신규 소스, 빌드 등을 배포 시부터 이력을 정확하게 관리하여 진행한다. 따라서 개발자와 테스터가 함께 협업을 해야 하고 최초 및 변경, 개선 항목까지 연결되어 다음 항목에 대한 정보들이 릴리즈 노트를 통해 작성되어야 한다.

1. 초기 버전 릴리즈 노트 작성 항목

릴리즈 노트는 보통 특정 소프트웨어 릴리즈의 최근 변경 사항, 개선 사항 및 버그 수정을 간결히 요약한 것이다. 사실 개발 조직 차원에서의 릴리즈 노트에 대한 표준 형식은 없다. 하지만 통상적으로 배포되는 정보의 유형과 사용자의 요구 사항에 기초하여 공통 항목으로 서식에 대한 다음 스타일은 정의되어야 한다.

〈표 1-1〉 릴리즈 노트 작성 항목

작 성 항 목	설 명
Header	문서 이름(릴리즈 노트 이름), 제품 이름, 버전 번호, 릴리즈 날짜, 참고 날짜, 노트 버전 등
개요	제품 및 변경에 대한 간략한 전반적 개요
목적	릴리즈 버전의 새로운 기능목록과 릴리즈 노트의 목적에 대한 간략한 개요. 버그 수정 및 새로운 기능 기술.
이슈 요약	버그의 간단한 설명 또는 릴리즈 추가 항목 요약
재현 항목	버그 발견에 따른 재현 단계 기술
수정/개선 내용	수정 / 개선의 간단한 설명 기술
사용자 영향도	버전 변경에 따른 최종 사용자 기준의 기능 및 응용 프로그램 상의 영향도 기술
SW 지원 영향도	버전 변경에 따른 SW의 지원 프로세스 및 영향도 기술
노트	SW 및 HW Install 항목, 제품, 문서를 포함한 업그레이드 항목 메모
면책 조항	회사 및 표준 제품과 관련된 메시지. 프리웨어, 불법 복제 방지, 중복 등 참조에 대한 고지 사항
연락 정보	사용자 지원 및 문의 관련한 연락처 정보

2. 릴리즈 노트 추가 작성 및 개선 사항 발생의 예외 케이스

릴리즈 정보의 예외 케이스 발생에 따른 추가 및 개선 항목이 나타날 수 있으므로 릴리즈 노트의 항목이 추가될 수 있다. 테스트 단계에서 베타 버전이 출시되거나, 긴급 버그 수정, 자체 기능 향상 등의 특이한 케이스 등이 발생할 수 있으므로 이러한 경우에도 추가 항목이 작성되어야 한다.

(1) 테스트 단계에서의 베타 버전 출시

제품 소프트웨어의 차기 버전이나 신규 버전의 베타 버전 테스트 단계에서도 릴리즈 버전으로 정보를 체크하여 릴리즈 노트를 작성할 수 있다. 이럴때는 자체에서 기준을 수립하여 현 베타 버전을 신규 소스로 하여 릴리즈를 할지, 예외 사항으로 베타 버전

에 대한 릴리즈 노트를 따로 만들지 사전에 정의해야 한다.

(2) 긴급 버그 수정 시

긴급한 버그가 발견되어 이를 수정할 경우의 릴리즈 노트 작성이다. 보통 긴급히 버그가 수정되면 릴리즈 노트 작성을 놓치는 경우가 많다. 반드시 버그 번호를 포함한 모든 수정된 버그를 기술하여 릴리즈 노트에 추가한다.

(3) 자체 기능 향상을 포함한 모든 추가 기능의 향상

자체적으로 기능 개선을 완료했을 때 정식으로 릴리즈 버전을 추가하고, 이에 따른 신규 릴리즈 노트를 작성한다. 업그레이드는 SW 및 HW에 대한 항목까지 포함된다.

(4) 사용자 요청에 따른 특이한 케이스 발생

제품 소프트웨어가 사용자에게 배포됨에 따라 기존에 배포된 릴리즈 노트의 연락처 정보를 통해 사용자의 의견이 접수된 경우이다. 개발 팀 내부에서 허용되는 범위 내에서 요청이 접수될 경우 이를 자체 기능 향상과는 별도의 버전으로 새로 추가하여 릴리즈 노트를 작성할 수 있다.

수행 내용 / 버전을 고려한 제품 릴리즈 노트 작성하기

재료 · 자료

- 신규 개발 소스
- 모듈 정의서, UI 정의서
- 사용자 환경과 맞는 최소 시스템 사양

기기(장비 · 공구)

- 컴퓨터, 인터넷
- 릴리즈 노트 작성용 Office, 모듈 빌드 프로그램

수행 순서



[그림 1-2] 릴리즈 노트 작성

① 신규 패키징 제품의 모듈, 빌드 내용을 식별한다.

1. 릴리즈 노트 작성을 위한 모듈 및 빌드 : 모듈 및 빌드 수행 후 릴리즈 노트 기준의 항목 순서대로 정리
2. I/O 데이터, 전체적인 기능 정의 및 데이터 흐름 : 소스 기능을 통해 처리되는 int, char 형 데이터, 기능 및 데이터의 흐름 정리
3. Function 단위 및 Output에 대한 상세 정의 : 메인 함수 이외의 호출 함수를 정의하고 이에 대한 출력값을 식별

② 패키징된 릴리즈 정보를 확인한다.

1. 문서 이름(릴리스 노트 이름), 제품 이름 : 릴리즈 노트 작성을 위해 문서명, 제품명의 릴리즈 기본 정보 확인
2. 버전 번호, 릴리즈 날짜 : 최초 패키징 버전 작성을 위한 번호, 초기 릴리즈 날짜 확인
3. 참고 날짜, 노트 버전 : 매번 패키징 수행 진행날짜 및 릴리즈 노트의 갱신 버전 확인

③ 빌드 내용에 따라 릴리즈 노트의 개요를 작성한다.

1. 제품 및 변경에 대한 간략한 전반적인 개요 : 빌드 이후에 제품 및 패키징에 대한 간략히 메모한다.
2. 개발 소스의 빌드에 따른 결과물 : 빌드 이후의 패키징본에 대한 결과를 기록한다.
3. 버전 및 형상 관리에 대한 전반적인 노트(Note) : 버전 번호 내용, 버전 관리 사항 등을 릴리즈 노트에 기록한다.

④ 이슈, 버그 및 추가 영향도를 체크하여 기술한다.

1. 버그의 간단한 설명 또는 릴리즈 추가 항목 : 발생한 버그의 설명, 개선한 릴리즈 항목을 기술한다.
2. 버그에 대한 내용 반드시 재현 및 재현 내용 : 버그 발견을 위한 재현 테스트 및 재현 환경을 기록한다.
3. S/W 및 사용자 입장에서 영향도 파악 : 이슈, 버그 발생에 따른 영향도를 상세 기술한다

⑤ 학습한 항목에 따른 내용을 포함하여 정식 릴리즈 노트를 작성한다.

1. 릴리즈 정보, Header 및 개요 반드시 포함 : 위의 내용을 포함하여 정식 릴리즈 노트에 기본 사항을 포함하여 기술한다.
2. 이번 정보의 릴리즈 노트 개요 : 정식 버전을 기준으로 릴리즈 노트의 개요를 작성한다
3. 이슈, 버그 등 개선에 따른 원인 재현 내용 : 개선한 포인트를 중심으로 원인, 테스트 내용을 기술한다.
4. 영향도 : 이슈, 버그에 따른 영향도를 상세히 기술한다.

⑥ 추가 개선에 따른 추가 항목을 식별하여 릴리즈 노트를 작성한다.

* 릴리즈 노트 양식은 <표 1-1> 상세 설명 참조.

1. 베타 버전 발생을 가정한 테스트 단계 : 추가 개선에 대한 베타 버전을 이용 테스트를 수행한다.
2. 긴급 버그 수정 : 테스트 중 발생한 긴급 버그를 즉시 수정한다.
3. 추가 기능 향상 : 버그 이외에 추가 기능 향상을 위해 작은 기능을 수정한다.
4. 사용자 요청에 따른 추가 개선 : 사용자 요청 발생 시 이에 따른 기능 개선을 계획 및 수정한다.

수행 tip

- 릴리즈 노트 작성 시, 기본적인 정식 릴리즈 노트에 대한 Header 정보 및 버전 개요를 우선 정의하고, 정식 노트를 작성하는 것이 필요하다. 그리고 이에 대한 정식 노트를 기준으로 추가, 개선 버전에 대해서 점차적으로 버전을 향상하여 릴리즈 노트를 작성해 나간다.

1-3. 패키징 도구를 활용한 설치, 배포 수행

학습 목표

- 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품 소프트웨어의 설치, 배포 파일을 생성할 수 있다.

필요 지식 /

① 제품 소프트웨어의 패키징 도구

제품 소프트웨어 패키징 도구는, 패키징 작업 진행 시에 암호화/보안 기능을 고려하여 패키징할 수 있도록 도와준다. 특히 디지털 콘텐츠의 불법 사용, 복제 방지, 과금 서비스를 통한 정상사 용자를 검증 가능한 저작권 기술을 감안한 도구를 통해 패키징하는 것이 중요하다.

1. 제품 소프트웨어 패키징 도구의 개념

제품 소프트웨어 패키징 도구는 배포를 위한 패키징 시에 디지털 콘텐츠의 지적 재산을 보호하고 관리하는 기능을 제공하며, 안전한 유통과 배포를 보장하는 도구이자 솔루션이다. 특히 패키징 도구는 불법 복제로부터 디지털 콘텐츠의 지적 재산을 보호해 주는 사용 권한 제어 기술, 패키징 기술, 라이선스 관리, 권한 통제 기술 등을 포함한다.

2. 패키징 도구 활용 시 고려 사항

(1) 반드시 암호화/보안을 고려한다.

패키징 시 사용자에게 배포되는 소프트웨어임을 감안하여 반드시 내부 콘텐츠에 대한 암호화 및 보안을 고려한다.

(2) 추가로 다양한 이기종 연동을 고려한다.

패키징 도구를 활용하여 여러 가지 이기종 콘텐츠 및 단말기 간 DRM 연동을 고려한다.

(3) 사용자 편의성을 위한 복잡성 및 비효율성 문제를 고려한다.

패키징 도구를 고려하면 사용자의 입장에서 불편해질 수 있는 문제를 고려하여, 최대한 효율적으로 적용될 수 있도록 한다.

(4) 제품 소프트웨어의 종류에 적합한 암호화 알고리즘을 적용한다.

암호화 알고리즘이 여러 가지 종류가 있는데, 제품 소프트웨어의 종류에 맞는 알고리즘을 선택하여 배포 시 범용성에 지장이 없도록 고려한다.

② 제품 소프트웨어 저작권 보호의 이해와 필요성

1. 저작권의 이해

문학 학술(學術) 또는 예술의 범위에 속하는 창작물인 저작물에 대한 배타적 독점적 권리로 타인의 침해를 받지 않을 고유한 권한

2. 저작권 보호 기술의 개념

콘텐츠 및 컴퓨터 프로그램과 같이 복제가 용이한 저작물에 대해 불법 복제 및 배포 등을 막기 위한 기술적인 방법을 통칭한다. 저작권 보호 기술은 다음과 같은 특성을 가진다.

- 콘텐츠 복제의 제한적 허용: 복제는 허용하나 사용자 확인을 거쳐 과금, 제품 소프트웨어 패키징 시에 사용자 확인에 대해 상용화 과금 정책 수립이 연계됨
- 종량제 BM을 갖는 제품 소프트웨어의 경우 요금 부과는 Clearing House를 통한 이용 시간에 비례한 과금(종량제 실현)
- 패키징 도구는 암호화/보안 등의 기능을 고려함. 따라서 패키징 제작자가 지정한 Business Rule과 암호가 함께 Packaging되어 배포

3. 저작권 보호 기술의 필요성

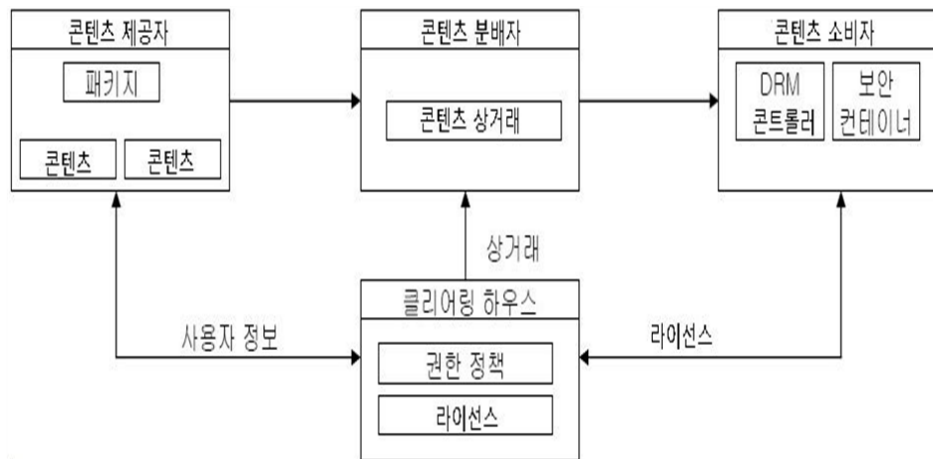
- 배포된 제품 소프트웨어의 무한 복제가 가능하고 원본과 복사본이 동일하게 배포될 특성을 가짐.
- 제품 소프트웨어가 상용으로 배포되면 다양한 부가 가치를 가질 수 있는 상품으로 바뀔때 따라 패키징 시에 원작자에 대한 권리 보호 우선 필요

③ 저작권 보호 측면의 패키징도구 활용

1. 패키징 수행과 디지털 저작권 관리(DRM)의 절차 및 흐름

(1) 저작권 관리의 흐름

콘텐츠 제공자, 분배자, 소비자 간의 패키징 배포 및 관리의 주체를 중앙의 클리어링 하우스에서 키 관리 및 라이선스 발급 관리



[그림 1-3] 저작권 관리 흐름도

<표 1-2> 저작권 관리 구성 요소

구성 요소	설 명
콘텐츠 제공자 (Contents Provider)	콘텐츠를 제공하는 저작권자
콘텐츠 분배자 (Contents Distributor)	쇼핑몰 등으로써 암호화된 콘텐츠 제공
패키저(Packager)	콘텐츠를 메타 데이터와 함께 배포 가능한 단위로 묶는 기능
보안 컨테이너	원본을 안전하게 유통 하기위한 전자적 보안 장치
DRM 컨트롤러	배포된 콘텐츠의 이용 권한을 통제
클리어링 하우스 (Clearing House)	키 관리 및 라이선스 발급 관리

2. 암호화/보안 기능 중심의 패키징도구 기술 및 활용

올바른 패키징 도구의 활용을 위해서는 암호화/보안 기능 중심의 요소 기술을 정확히 이해하고 있어야 한다. 이 기술을 바탕으로 패키징 작업 시에 도구를 이용하여 나오는 결과물에 대해서도 신뢰할 수 있도록 작업이 되어야 한다.

<표 1-3> 패키징 도구 구성 요소

구성 요소	설 명
암호화(Encryption)	<ul style="list-style-type: none"> - 콘텐츠 및 라이선스를 암호화하고, 전자 서명을 할 수 있는 기술 - PKI, Symmetric/Asymmetric Encryption, DiGital Sinature
키 관리 (Key Manangement)	<ul style="list-style-type: none"> - 콘텐츠를 암호화한 키에 대한 저장 및 배포 기술 (Centralized, Enveloping)
암호화 파일 생성 (Packager)	<ul style="list-style-type: none"> - 콘텐츠를 암호화된 콘텐츠로 생성하기 위한 기술 - Pre-packaging, On-the-fly Packaging
식별 기술 (Identification)	<ul style="list-style-type: none"> - 콘텐츠에 대한 식별 체계 표현 기술 - DOI, URI
저작권 표현 (Right Expression)	<ul style="list-style-type: none"> - 라이선스의 내용 표현 기술 - XrML/MPGE-21 REL, ODRL
정책 관리 (Policy management)	<ul style="list-style-type: none"> - 라이선스 발급 및 사용에 대한 정책표현 및 관리기술 - XML, Contents Management System
크랙 방지 (Tamper Resistance)	<ul style="list-style-type: none"> - 크랙에 의한 콘텐츠 사용 방지 기술 - Code Obfuscation, Kernel Debugger Detection, Module Certification - Secure DB, Secure Time Management, Encryption
인증(Authentication)	<ul style="list-style-type: none"> - 라이선스 발급 및 사용의 기준이 되는 사용자 인증 기술 - User/Device Authentication, SSO, DiGital Certificate

- 패키징 도구 적용 관련 기술적 동향

- (1) 패키징 도구 벤더에 의해 개발되던 각종 기술들이 표준화를 통해 통합 플랫폼화
- (2) 클라우드 환경에서 디지털 콘텐츠의 투명한 접속, 이용, 이동 등의 사용 편리성 보장
- (3) 특정 도구나 환경에서만 적용되지 않는 범용성 확보, 벤더 간 식별 체계, 상호 호환성 표준화 노력

수행 내용 / 패키징 도구를 활용한 설치, 배포 수행하기

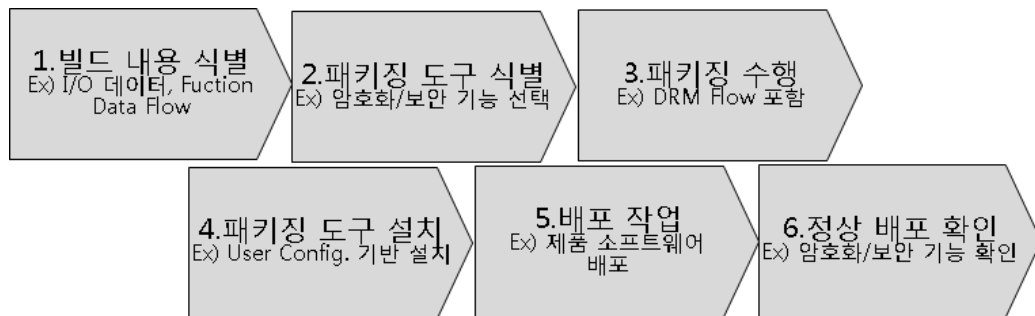
재료 · 자료

- 신규 개발 소스
- 사용자 환경과 맞는 최소 시스템 사양
- 패키징 도구

기기(장비 · 공구)

- 컴퓨터, 인터넷
- 모듈 빌드 프로그램

수행 순서



[그림 1-4] 설치, 배포 수행순서

① 신규 패키징 제품의 모듈, 빌드 내용을 식별한다.

1. 릴리즈 노트 작성을 위한 모듈 및 빌드 : 모듈화, 패키징을 통한 릴리즈 노트의 항목을 정리한다.
2. I/O 데이터, 전체적인 기능 정의 및 데이터 흐름 : 소스 기능을 통해 처리되는 int, char 형 데이터, 기능 및 데이터의 흐름을 정리한다.
3. Function 단위 및 Output에 대한 상세 정의 : 메인 함수 이외의 호출 함수를 정의하고, 이에 대한 출력값을 식별한다.

② 암호화/보안 중심의 패키징 도구를 식별한다.

1. 패키징 도구의 사전 선택 : 패키징 도구 선택 시 암호화/보안 기능을 고려한다.
2. 암호 /보안 기능 확인 : 패키징 도구에서 암호화/보안의 기능을 식별한다.

③ DRM 흐름을 확인하여 패키징을 수행한다.

1. 콘텐츠 분배자, 배포자, 소비자 간의 DRM 흐름을 확인 : DRM 흐름도를 참고하여 3자간의 흐름을 확인한다.
2. 패키징 수행 시 키 관리, 보안 개념 확인하며 수행 : 반드시 키 관리, 보안 개념을 반영한 패키징을 수행한다.

④ 검토된 패키징 도구를 설치 작업을 진행한다.

1. 환경에 맞게 패키징 도구 설치 작업 진행 : 사전 선택된 패키징 도구 설치 진행한다.
2. 패키징 도구 설치 완료 후 정상 동작 확인 : 인스톨 이후에 패키징 도구의 여러 메뉴 및 기능을 실행한다.

⑤ 검토된 패키징 도구를 활용하여 배포 작업을 진행한다.

1. 패키징 도구 설치 이후 제품 소프트웨어의 배포 작업을 진행 : 도구 설치 이후 정식 패키징 작업을 수행한다.
2. 배포 후 최종 패키징 완료 확인 : 패키징 도구를 통한 정식 패키징 수행, 이후 결과를 확인한다.

⑥ 최종 패키징 완료 후 암호화/보안 기능이 정상적으로 적용되었는지 확인한다.

1. 암호화/보안 기능 적용 확인 : 패키징 도구의 암호화/보안의 기능이 적용 되었는지 체크한다.
2. 제품 소프트웨어 배포본 기준으로 암호화/보안 기능 체크리스트 확인 : 빌드 도구를 통해 패키징을 수행하고, 최종본에서 암호화/보안을 중심으로 체크리스트를 확인한다.

수행 tip

- 패키징 도구 활용 시, 작업 진행 시에 암호화/보안 기능을 고려하여 패키징할 수 있도록 하는 것이 가장 중요하다. 불법 사용 방지, 복제 방지, 과금 기술 등의 주요 요소 기술들을 숙지하여 패키징 결과에 신뢰성을 확보할 수 있도록 한다.

학습 1 교수 · 학습 방법

교수 방법

- 교수자의 주도로 샘플 버전의 신규 소스를 설명하고, 패키징 활용을 위한 패키징 도구를 설명한다.
- 교수자의 주도로 샘플 버전의 신규 소스를 설명하고, 릴리즈 노트 작성을 위한 패키징 개요를 설명한다.
- 릴리즈 노트 작성을 위해 Header 정보를 공유한다.
- 릴리즈 노트의 개요 및 목적 등을 설명하고, 노트 작성을 가이드한다.
- 릴리즈 노트의 변경, 개선을 위한 이슈, 버그 및 추가 영향도를 제시한다.
- 이슈, 버그 및 추가 영향도에 따른 추가 개선 버전의 릴리즈 노트를 작성하도록 하고, 구두 발표를 진행한다.
- 암호화/보안의 중요성을 설명하고, 패키징 도구의 선택방향을 가이드한다.
- DRM의 개념 및 저작권 보호에 따른 패키징 방향을 제시한다.
- 패키징 도구를 활용 시 효과적으로 암호화/보안을 수행할 수 있도록 요소 기술을 설명한다.
- 패키징 도구를 활용하여 설치 및 배포 작업을 하고, 결과를 피드백할 수 있도록 한다.

학습 방법

- 신규 소스 및 릴리즈 노트 작성을 위해 소스의 기능, 패키징 도구의 개요를 이해한다.
- Header의 항목에 따라 릴리즈 정보를 식별한다.
- 릴리즈 노트의 개요 및 목적을 통해 릴리즈 노트를 작성할 수 있다.
- 릴리즈 버전의 추가 개선을 통해 대응 방향 및 영향도 파악을 진행할 수 있다.
- 개선 버전의 릴리즈 노트를 추가 작성하고, 이에 따른 릴리즈 버전 관리를 할 수 있다.
- 패키징 도구에서의 암호화/보안을 학습한다.
- DRM의 흐름 및 저작권 보호에 대한 개념을 이해한다.
- 패키징 도구에 적용된 주요 요소 기술을 학습한다.
- 패키징 도구를 통해 작업된 제품 소프트웨어에 대한 저작권 진단을 하고 체크리스트를 작성한다.

학습 1 평 가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
사용자 중심의 패키징 수행	- 신규 개발, 변경, 개선된 제품 소프트웨어의 소스들로부터 모듈들을 빌드하고, 고객의 편의성을 고려하여 패키징 할 수 있다.			
버전을 고려한 제품 릴리즈 노트 작성	- 이전 릴리즈 이후의 변경, 개선 사항을 포함하여 신규 패키징한 제품 소프트웨어에 대한 릴리즈 노트를 작성할 수 있다.			
패키징 도구를 활용한 설치, 배포 수행	- 저작권 보호를 위해 암호화/보안 기능을 제공하는 패키징 도구를 활용하여, 제품 소프트웨어의 설치, 배포 파일을 생성할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
사용자 중심의 패키징 수행	- 신규 개발, 변경, 개선된 제품 소프트웨어 소스들의 기본 모듈 및 기능 식별 능력 - 모듈 단위로 분류 및 빌드를 수행하는 능력 - 고객의 편의성을 고려한 사용자 환경 검토 능력			
버전을 고려한 제품 릴리즈 노트 작성	- 이전 릴리즈 및 이후의 변경, 개선 사항 분류 능력 - 신규 패키징한 제품 소프트웨어에 대한 릴리즈 노트의 개요 및 Layout 작성 능력 - 정식 릴리즈 노트에 패키징 이후 추가 개선 사항의 반영 능력			
패키징 도구를 활용한 설치, 배포 수행	- 저작권 보호를 위한 암호화/보안 기능 숙지 수준 - 패키징 도구 사용법 및 활용 능력 - 패키징 도구를 활용한 제품 소프트웨어의 설치, 배포 능력			

• 구두 발표

학습 내용	평가 항목	성취수준		
		상	중	하
사용자 중심의 패키징 수행	<ul style="list-style-type: none"> - 신규 개발, 변경, 개선된 제품 소프트웨어 소스들의 기본 모듈 및 기능 식별 능력 - 모듈 단위로 분류 및 빌드를 수행하는 능력 - 고객의 편의성을 고려한 사용자 환경 검토 능력 			
버전을 고려한 제품 릴리즈 노트 작성	<ul style="list-style-type: none"> - 이전 릴리즈 및 이후의 변경, 개선 사항 분류 능력 - 신규 패키징한 제품 소프트웨어에 대한 릴리즈 노트의 개요 및 Layout 작성 능력 - 정식 릴리즈 노트에 패키징 이후 추가 개선 사항의 반영 능력 			
패키징 도구를 활용한 설치, 배포	<ul style="list-style-type: none"> - 저작권 보호를 위한 암호화/보안 기능 숙지 수준 - 패키징 도구 사용법 및 활용 능력 - 패키징 도구를 활용한 제품 소프트웨어의 설치, 배포 능력 			

피드백

1. 핵심 용어 및 학습 목표 피드백

- 제품 소프트웨어 모듈화 및 기본 모듈의 식별 능력을 숙지한다.
- 릴리즈 버전을 고려한 릴리즈 노트 작성 능력을 향상시킬 수 있도록 연습한다.
- 패키징 도구를 활용한 DRM 및 저작권 개념 확립을 우선 정리한다.
- 패키징 도구를 통한 제품 소프트웨어 설치 및 배포 수행을 실습한다.

2. 평가 항목 피드백

- 제품 소프트웨어의 소스들의 기본 모듈 및 기능을 향상시킨다.
- 모듈 단위로 분류 및 빌드를 수행하는 능력을 숙달한다.
- 고객의 편의성을 고려한 사용자 환경 검토 능력을 향상시킨다.
- 이전 릴리즈 및 이후의 변경, 개선 사항 분류해 낼 수 있도록 한다.
- 신규 패키징한 제품 소프트웨어에 대한 릴리즈 노트의 개요 및 Layout 작성 능력을 향상시킨다.
- 릴리즈 노트에 패키징 이후 추가 개선 사항을 정식 릴리즈 노트에 반영, 작성하는 절차를 숙달한다.
- 저작권 보호를 위한 암호화/보안의 기술적 개념을 숙지한다.
- 패키징 도구의 사용법 및 활용 방법을 숙지한다.
- 패키징 도구를 활용한, 제품 소프트웨어의 설치, 배포 능력을 향상시킨다.

3. 구두 발표를 통한 피드백 공유

- 발표한 내용 중 미비 사항이나 보완해야 할 사항을 정리하여 돌려준다.

학습 1	제품 소프트웨어 패키징하기(LM2001020209_14v2.1)
학습 2	제품 소프트웨어 매뉴얼 작성하기 (LM2001020209_14v2.2)
학습 3	제품 소프트웨어 버전 관리하기(LM2001020209_14v2.3)

2-1. 제품 소프트웨어 설치 매뉴얼 구성 수립 및 작성

학습 목표

- 사용자가 제품 소프트웨어를 설치하는 데 참조할 수 있도록 제품 소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.

필요 지식 /

① 제품 소프트웨어 설치 매뉴얼의 개요

1. 제품 소프트웨어의 매뉴얼의 파악

제품 소프트웨어의 매뉴얼은 제품 소프트웨어 개발단계부터 적용한 기준이나 패키징이 후 설치 및 사용자 측면의 주요 내용 등을 문서로 기록한 것이며, 사용자 중심의 기능 및 방법을 나타낸 설명서와 안내서를 의미한다.

실제 개발자들이 많이 겪어 보지 못하는 영역이어서 개발보다도 더 어려움을 겪는 작업이다. 요즘은 매뉴얼 작성 프로그램이나 툴을 이용하기도 하지만 전통적으로 작성 내용에 맞추어 항목별로 작성하게 된다.

2. 설치 매뉴얼 작성의 기본 사항

- (1) 설치 매뉴얼은 개발자의 기준이 아닌 사용자의 기준으로 작성한다.
- (2) 최초 설치 실행부터 완료까지 순차적으로 진행한다.
- (3) 각 단계별 메시지 및 해당 화면을 순서대로 전부 캡처하여 설명한다.
- (4) 설치 중간에 이상 발생 시 해당 메시지 및 에러에 대한 내용을 분류하여 설명한다.

② 제품 소프트웨어 설치 매뉴얼의 작성 항목

제품 소프트웨어 설치 매뉴얼은 기본적인 개요 및 서문 이외에도 상세한 설치 항목에 대해서 순서대로 설명해 가는 것이 중요하다.

1. 제품 소프트웨어 설치 매뉴얼의 기본 작성 항목

(1) 목차 및 개요

(가) 목차

매뉴얼 전체의 내용을 순서대로 요약

(나) 개요

- 설치 매뉴얼의 주요 특징에 대해 정리
- 설치 매뉴얼에서의 구성과 설치 방법, 순서 등에 대해 기술함

(2) 서문

(가) 문서 이력 정보

<표 2-1> 문서 이력 정보

버전	작성자	작성일	검토자	일시	검수인
v0.1	홍길동	2015-10-01			
변경 내용	최초 작성				

(나) 설치 매뉴얼의 주석

- 주의 사항: 사용자가 제품 설치 시 반드시 숙지해야 하는 중요한 정보 주석 표시
- 참고 사항: 설치 관련하여 영향을 미치는 특별한 사용자 환경 및 상황에 대한 내용 주석 표시

(다) 설치 도구의 구성

- 프로그램 파일 구성(설치 파일 및 폴더)
 - exe / dll / ini / chm 등 해당 설치 관련 파일 설명
- 폴더 및 설치 프로그램 실행 파일 설명
- 설치 이후에 설치 결과를 기록하는 Log 폴더 설명

제품 설치와 관련하여 아래의 환경 체크 항목들을 공지한다.

<표 2-2> 환경 체크 항목

확인 항목	체크할 내용
사용자 환경	사용자의 CPU 및 Memory, OS 등의 적합 환경
응용 프로그램	설치 전 다른 응용 프로그램의 종료
업그레이드 버전	업그레이드 이전 버전에 대한 존재 유무 확인
백업 폴더 확인	데이터 저장 폴더를 확인하여 설치 시 폴더 동기화

(3) 기본 사항

기본적으로 제품 소프트웨어 관련하여 설명되어야 할 항목들은 다음과 같다.

이때 간략한 기능 및 UI를 첨부하고, 요약한다.

<표 2-3> 매뉴얼 기본 사항

기본 사항	항목 설명
제품 소프트웨어 개요	- 제품 소프트웨어의 주요 기능 및 UI 설명 - UI 및 화면 상의 버튼, 프레임 등을 도식화하여 설명
설치 관련 파일	- 제품 소프트웨어를 설치하기 위한 관련 파일 설명 - 설치 구동을 위한 exe 실행 - ini나 log 파일 같은 관련 파일
설치 아이콘(Installation)	Windows 구동용 설치 아이콘 설명
프로그램 삭제	해당 소프트웨어 삭제 시 원래대로 삭제하는 방법을 설명
관련 추가 정보	- 제품 소프트웨어 이외의 관련 설치 프로그램 정보(Build) - 관련 프로그램 제작사 추가 정보 기술

2. 제품 소프트웨어 설치 매뉴얼 작성 방법 상세 가이드

(1) 설치 매뉴얼 내용

제품 소프트웨어의 설치 방법을 순서대로 상세하게 설명한다. 주로 화면을 계속 캡처하여 사용자가 이해하기 쉽도록 구성한다.

(가) 설치 화면 및 UI

- 설치 실행: exe 등의 Install 파일을 실행하도록 화면 첨부
메인 화면 및 안내창: 이미지로 해당 내용을 직접 첨부하여 설치 시 나타나는 내용 설명
- 표시되는 내용 및 안내창의 메시지 설명

(나) 설치 이상 시 메시지 설명

- 설치가 잘못되었거나 잘못된 환경일 경우 메시지

- 해당 이상 내용에 따른 메시지를 설명

ex) 시스템 이상/ xx 파일 오류 등등 으로 해당 항목에 대한 설치가 불가능합니다.

1. 해당 원인 및 메시지 설명 / 오류 코드 등을 표로 첨부
2. 해당 화면 캡처 첨부

- 설치 단계별 참고 사항, 주의 사항 등을 메모로 추가

(다) 설치 완료 및 결과

- 정상 설치 완료 시 최종 메시지 출력
- 정상 결과 및 사용에 대한 안내를 함으로써 설치가 잘되었음을 최종 통지

(라) FAQ

- 실제 설치 시 자주 발생하는 어려움들을 FAQ로 정리하여 제품이 설치되지 않을 때를 대비한다.

(마) 설치 시 점검 사항

- 설치 전 사용자 환경에 맞추어 점검할 내용들을 설명한다.
- 사용자 계정 및 설치 권한에 대해 확인할 수 있도록 한다.
- 설치 중간에 에러 발생 시 체크할 사항을 설명한다.

(바) Newwork 환경 및 보안

- 설치 시 Newwork 문제로 인해 오류가 발생하지 않도록 사전에 Connection을 체크한다.
- 보안 및 방화벽 등의 문제로 설치가 안될 경우에 대비해 환경을 체크할 수 있도록 가이드한다.

(2) 고객 지원 방법(Customer Support)

- 유선 및 E-mail, Website URL
- 설치 관련하여 기술적인 지원이나 제품 서비스를 원하시면, 아래의 표를 참조하거나 해당 구입사에 문의하십시오 .

주의: 요청 시 제품 번호가 필요합니다.

<표 2-4> 고객 지원 방법

국가	Website	전화	Email
한국	www.Website.com	02)123-4567	support@Website.com

(3) 준수 정보 & 제한 보증(Compliance Information & Limited Warranty)

- Serial 보증, 불법 등록 사용 금지 등의 준수 사항 권고
- 저작권 정보 관련 작성(출처: Android, Microsoft)

1. 저작권자 소유권 정보

Copyright © 제품 소프트웨어 ® Corporation, 2015. 모든 권리는 저작권자의 소유입니다.

2. SW 허가권 관련 명시

Android는 Google Inc의 등록 상표입니다. 이 상표 사용 허가권은 Google에 있습니다.

Google TM, Google 로고, Android TM, Android 로고 및 Gmail TM은 Google Inc의 상표입니다. 이 상표의 사용 허가권은 Google에 있습니다.

Microsoft, MS, Windows, MS Office는 미국, 대한민국, 및/또는 기타 국가에서 Microsoft Corporation의 등록 상표 또는 상표입니다.

여기서 언급된 기타 다른 제품과 회사 이름은 각각 그 소유자의 상표일 수 있습니다.

3. 확장 플랫폼, 응용 Application

본 제품은 다양한 기반 응용 프로그램 소프트웨어에 의해 확장될 수 있는 Linux에 기반을 둔 Android 플랫폼을 탑재하고 있습니다.

4. 통신 규격

Wi-Fi 및 Wi-Fi CERTIFIED 로고는 Wireless Fidelity Alliance에 등록된 상표입니다. Bluetooth 및 Bluetooth 로고는 Bluetooth SIG, Inc가 소유한 상표입니다.

5. 저장 장치

microSD TM은 SD Card Association의 상표입니다.

6. 개발 언어

Java, JME 및 기타 Java 기반 마크는 미국 및 기타 국가에서 Sun Microsystems, Inc의 상표 또는 등록 상표입니다.

7. 연동 프로그램

연동 프로그램은 ABC, Inc의 등록 상표입니다. All rights reserved(모든 권리 보유). 연동 프로그램은 ABC, Inc의 상표 또는 등록 상표입니다. All rights reserved(모든 권리 보유).

8. 문서 효력

본 문서의 어떠한 부분도 ViewSonic Corporation의 사전 서면 허가 없이 어떠한 목적 및 수단으로든 복사, 재 생산, 또는 전달될 수 없습니다.

9. 지적 소유권 정보

본 제품에 속하는 모든 기술 및 제품에 대한 권리는 관련 소유자의 재산입니다.

수행 내용 / 제품 소프트웨어 설치 매뉴얼 작성하기

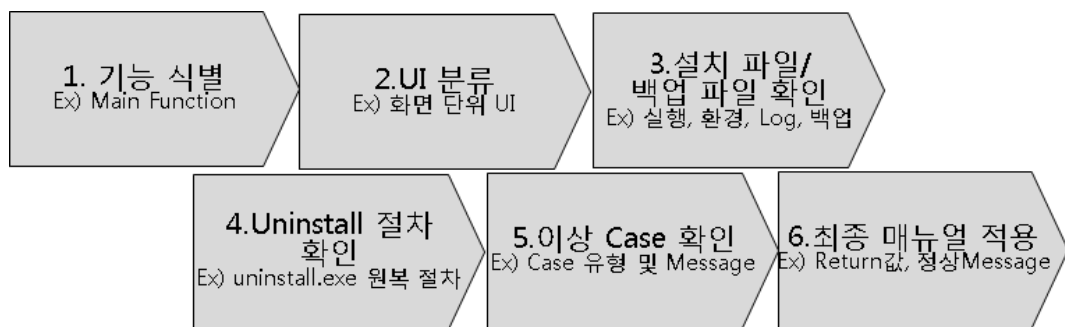
재료 · 자료

- 제품 소프트웨어 샘플
- 제품 기능 정의서, UI 정의서
- 사용자 환경과 맞는 최소 시스템 사양

기기(장비 · 공구)

- 컴퓨터, 인터넷
- Office, Document 구동 프로그램

수행 순서



[그림 2-1] 설치매뉴얼 수행순서(자체 제작)

① 제품 소프트웨어의 개요 및 기능들을 식별한다.

1. 제품 소프트웨어 개발 목적 및 설명

제품 소프트웨어 자체의 목적 및 내용을 설명할 수 있도록 한다.

2. 제품 소프트웨어의 주요 기능적 설명

제품의 전체적인 기능을 흐름순으로 정리하여 설명한다.

② 화면 단위로 작성될 UI를 분류한다.

1. 설치를 위한 화면 단위 및 메뉴 분류

화면 중심으로 UI와 메뉴를 순서대로 분류한다.

2. UI 정의서에 기초한 메인 항목 분류

사전에 작성된 UI 정의서를 항목별로 분류한다.

3. 설치 매뉴얼에 작성될 순서대로 UI를 분류

설치 매뉴얼 기준의 양식을 목차에 맞게 UI를 분류한다.

③ 설치 파일 및 백업 파일 등 관련 파일을 확인한다.

1. 설치 파일 및 백업 파일명 확인

실제로 제품을 설치할 파일(exe) 및 백업 파일명을 확인한다.

2. 설치 파일 및 백업 파일명 폴더 위치

실제로 제품을 설치할 파일(exe) 및 백업 파일명의 폴더 위치를 확인한다.

3. 파일별 기능(실행, 환경, Log, 백업)

실행, 환경, Log, 백업 등의 다양한 파일 들을 확인하고 기능을 숙지한다.

④ 추후 Uninstal을 위한 절차를 확인한다.

1. Uninstal 진행 시 실행 파일

제품의 제거시를 고려하여 Uninstal 파일을 설명한다.

2. Uninstal 진행 시 단계

제품의 제거 시의 단계를 순서대로 기술한다.

3. Uninstal 이후 정상 상태로의 원복

Uninstal 이후 설치 전 상태로의 원복을 최종 확인한다.

⑤ 설치 진행 시 이상 메시지 등에 대한 Case를 발생 시키고, 확인한다.

1. 설치에 대한 이상 Case Test

설치 시 발생하는 이상 현상을 발생하고 관련 테스트를 수행한다.

2. 다양한 Case명 발생 메시지

다양한 이상 현상 발생 시 이에 따른 메시지를 정리한다.

3. Case별로 발생하는 메시지가 정상적인지 확인

다양한 이상 현상의 내용에 맞는 메시지가 간결하고 정상적으로 표시되는지 확인한다.

⑥ 최종 설치 정상 완료 시 결과를 캡처후 최종 매뉴얼에 적용한다.

1. 설치 최종 완료 시 결과 확인

설치 최종 완료 후 이에 대한 메시지 및 최종 결과를 확인한다.

2. 결과에 대한 화면 및 해당 메시지 캡처

최종 설치 정상 완료를 확인하고 이에 대한 메시지를 캡처한다.

3. 최종 매뉴얼을 확인하고 마지막에 정상 결과 적용

최종 매뉴얼을 정리 검토하고 최종적으로 정상 결과를 적용한다.

수행 tip

- 설치 매뉴얼 작업 시 전체적인 기능이 화면 및 UI별로 순서대로 진행되는지가 중요하다. 가장 기본이 되어야 할 고객 편의성 측면과 이에 따른 절차적 설명을 도출해 내는 것이 중요한 포인트이다.

2-2. 제품 소프트웨어 사용자 매뉴얼 구성 수립 및 작성

학습 목표

- 사용자가 제품 소프트웨어를 사용하는 데 참조할 수 있도록 제품 소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.

필요 지식 /

① 제품 소프트웨어 사용자 매뉴얼의 개요

제품 소프트웨어 사용자 매뉴얼은 사용자 관점으로 진행되며, 이를 위한 제품 기능 및 고객 지원까지를 포함한다. 따라서 제품 소프트웨어의 특성을 먼저 이해하고 작성을 진행한다.

1. 사용자 매뉴얼 작성의 기본 사항

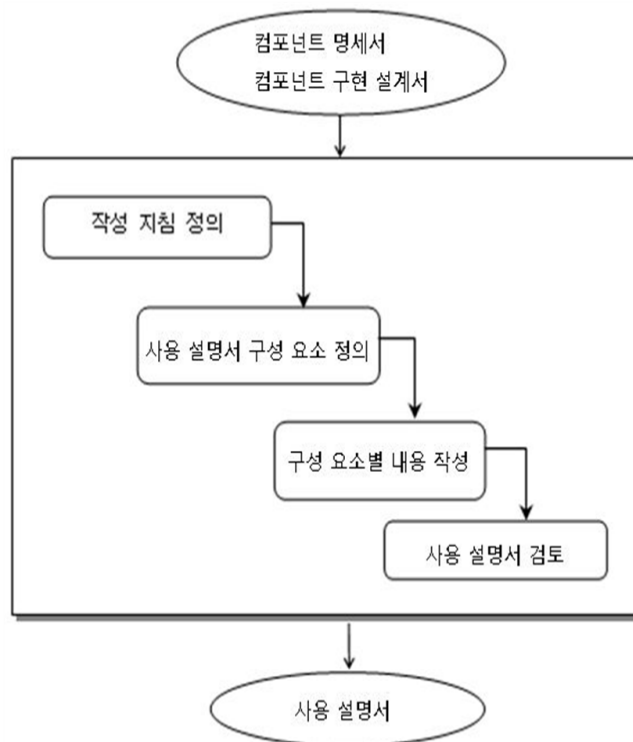
제품 소프트웨어 사용자 매뉴얼은 개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼을 작성하며, 제품 소프트웨어에 대한 패치 개발과 업그레이드를 위해 버전 관리를 수행한다. 개발된 컴포넌트 사용 시에 알아야 할 내용을 기술하며 패키지의 기능, 패키지의 인터페이스, 포함하고 있는 메서드나 오퍼레이션과 메서드의 파라미터 등의 설명이 포함되어 있다.

<표 2-5> 사용자 매뉴얼 작성 단계

작성 단계	단계별 설명
(1) 작성 지침 정의	사용자 매뉴얼을 작성하기 위한 지침을 설정한다. 사용자 매뉴얼은 실제 사용자 환경에 필요한 정보를 제공할 수 있는 형태로 작성될 수 있도록 하여야 한다.
(2) 사용자 매뉴얼 구성요소 정의	<ul style="list-style-type: none"> - 제품 소프트웨어의 기능 - 구성 객체 목록 - 객체별 메서드 - 메서드의 파라미터 및 설명 - 실제 사용 예제 - 사용자 환경 세팅 방법
(3) 구성 요소별 내용 작성	제품 소프트웨어 구성 요소별로 내용을 작성한다.
(4) 사용자 매뉴얼 검토	작성된 사용자 매뉴얼이 개발된 제품의 기능을 제대로 설명하는지,

작성 단계	단계별 설명
	<p>제품 사용 시 모자라는 정보가 없는지 등을 검사한다. 해당 기능별 관련 개발자와 함께 검토하면 기능 내용이나 인터페이스, 메서드나 메서드의 파라미터 등이 보다 정확히 반영할 수 있어서 더욱 효과적이다.</p> <ul style="list-style-type: none"> * 개발된 프로그램을 사용자 지침서의 내용에 따라 수행 시킨다. * 프로그램과 맞지 않는 부분이 있는지를 점검한다. * 점검 사항을 반영하여 사용자 지침서를 수정, 보완한다.

사용자 매뉴얼 작성은 명세서나 구현 설계서로부터 다음과 같은 단계를 거쳐서 최종 작성된다.



[그림 2-2] 사용자 매뉴얼 작성 절차

2. 사용자 매뉴얼과 사용자 환경

- (1) 제품 소프트웨어는 개발자가 아닌 사용자 중심으로 진행된다.
- (2) 신규 및 변경 개발 소스를 식별하고, 이를 모듈화하여 상용 제품으로 패키징한다.
- (3) 고객의 편의성을 위해, 신규/변경 이력을 확인하고, 이를 버전 관리 및 릴리즈 노트를 통해 지속적으로 관리해 간다.
- (4) 사용자의 실행 환경을 이해하고, 범용 환경에서 사용이 가능하도록 일반적인 배포 형태로 분류하여 패키징이 진행된다.

② 제품 소프트웨어 사용자 매뉴얼 작성 항목

제품 소프트웨어 사용자 매뉴얼은 실제 사용자를 위한 전체적인 서문 및 사용자 환경, 사용법을 위주로 작성되는 것이 일반적이다.

1. 제품 소프트웨어 사용자 매뉴얼의 기본 작성 항목

(1) 목차 및 개요

(가) 목차

매뉴얼 전체의 내용을 순서대로 요약

(나) 개요

- 제품 소프트웨어의 주요 특징에 대해 정리
ex) Windows 기반의 PC에서 제공하는 운영 체제와 SW의 중요한 특징에 대한 여러 기능을 갖는 소프트웨어임.
- 사용자 매뉴얼에서의 구성과 실행 방법, 메뉴에 대한 설명을 비롯하여 사용법, 각 항목에 따른 점검 기준, 그리고 설정 방법 등에 대해 기술함

(2) 서문

(가) 문서 이력 정보

<표 2-6> 사용자 매뉴얼 문서 이력 정보

버전	작성자	작성일	검토자	일시	검수인
v0.1	홍길동	2015-10-01			
변경 내용	최초 작성				

(나) 사용자 매뉴얼의 주석

- 주의 사항: 사용자가 반드시 숙지해야 하는 중요한 정보의 주석 표시
- 참고 사항: 특별한 사용자 환경 및 상황에 대한 내용의 주석 표시

(다) 기록 보관

- 제품 등록과 관련한 기록에 대한 내용 기재
- 향후 필요한 경우나 도움이 되는 추가 제품 정보를 받기 위한 내용 기재
- 인터넷을 이용한 지원이 가능하도록 Website URL 기재
- 제품 소프트웨어 CD-ROM 등을 통해 우편 또는 팩스로 지원을 보낼 수 있는 양식 제공

제품 등록과 관련하여 아래의 기록 항목들을 기재한다.

<표 2-7> 사용자 매뉴얼 기록 항목

매뉴얼 기록 항목	기재할 내용
제품 명칭	제품 소프트웨어의 명칭 기재
제품 소프트웨어의 모델명 (버전명)	버전을 포함한 제품 소프트웨어의 자체 모델명을 기재
기록 항목에 대한 문서 번호	제품 소프트웨어 Rev. 버전 날짜 등의 고유한 문서번호 기재
제품 번호	고유한 제품의 시리얼 넘버(Serial Number) 기재
구입 날짜	사용자가 직접 기재할 수 있도록 빈 공란으로 공백 처리함

(3) 기본 사항

기본적으로 제품 소프트웨어와 관련하여 설명할 항목들은 다음과 같다.

이때 간략한 기능 및 UI를 첨부하고, 요약한다.

<표 2-8> 사용자 매뉴얼 기본 사항

기본 사항	항목 설명
제품 소프트웨어 개요	<ul style="list-style-type: none"> - 제품 소프트웨어의 주요 기능 및 UI 설명 - UI 및 화면 상의 버튼, 프레임 등을 도식화하여 설명
제품 소프트웨어 사용	<ul style="list-style-type: none"> - 제품 소프트웨어를 사용하기 위한 최소 환경 설명 - PC 사양(CPU, Memory 등), OS 버전 명시 - 제품 소프트웨어 최초 동작을 위한 설명(실행 파일 or Website URL 등) - 주의 사항: 제품 소프트웨어 동작 시 프로그램 충돌사항 혹은 안전하게 사용하기 위한 주의 사항(개인정보, 보안 등에 관련된 사항까지 명시)
제품 소프트웨어 관리	제품 소프트웨어의 사용 종료 및 관리 등에 대한 내용 기재
모델, 버전별 특징	제품 구별을 위한 모델, 버전별 UI 및 기능의 차이 간단히 기술
기능, I/F의 특징	제품의 기능 및 Interface의 특징을 간단히 기술
제품 소프트웨어 구동 환경	<ul style="list-style-type: none"> - 개발 언어 및 호환 OS - 설치 마법사(Setup Wizard) 이후 사용자가 구동하기 까지의 과정 요약(Windows, Linux)

2. 제품 소프트웨어 사용자 매뉴얼 작성 방법 상세 가이드

(1) 사용자 매뉴얼 내용

제품 소프트웨어의 사용 방법을 다양한 측면별로 나누어 상세하게 설명한다. 주로 화면을 계속 캡처하여 사용자가 이해하기 쉽도록 구성한다.

(가) 사용자 화면 및 UI

- 주의 사항: 사용자가 반드시 숙지해야 하는 중요한 정보의 주석 표시
- 참고 사항: 특별한 사용자 환경 및 상황에 대한 내용의 주석 표시

(나) 주요 기능 분류

- 설명할 기능을 포함할 화면을 첨부
- 동작하는 기능을 화면의 순서대로 차례로 분류

ex) 기능별로 동작이나 해당 화면명을 기술하며 설명하면 이해가 쉽다.

1. A기능: XX하는 동작입니다.
2. B기능: 버튼명 / 기능명 등등 표시

- 기능 동작 시 참고 사항, 주의 사항 등을 메모로 추가

(다) 응용 프로그램(Programs)/설정(Settings)

- 제품 소프트웨어 동작 시 함께 동작하는 Application이나 설치되어 충돌될 수 있는 응용 프로그램에 대해 보여 준다.
- 제품 소프트웨어 동작 시 사전에 실행해야 할 Application이 있으면 기술한다.
- 제품 소프트웨어의 설정(Settings)과 관련한 사항이나 기본값에 대한 내용을 설명한다.

(라) 장치 연동

- 제품 소프트웨어가 Embedded 관련된 제품일 경우에 해당 Device에는 어떤 것이 있는지, 연동되는 Device에는 무엇이 있는지 설명한다.

(마) Network 환경

- 제품 소프트웨어와 관련한 Network 정보 표시(Status)
- Network에 정상 연결되었는지, 이를 위한 관련 설정값은 무엇이 있는지 기술

(바) Profile 설명

- 제품 소프트웨어 구동 시 체크하는 환경 파일
- 환경 파일의 경로 변경, 이동 금지
- 필수 파일의 내용 간략히 설명

(2) 액세서리(Accessory): SW Toolkit, Device

(3) 고객 지원 방법(Customer Support)

유선 및 E-mail, Website URL

기술적인 지원이나 제품 서비스를 원하시면, 아래의 표를 참조하거나 해당 구입 사에 문의하십시오 .

주의: 요청 시 제품 번호가 필요합니다.

<표 2-9> 고객 지원 방법

국가	Website	전화	Email
한국	www.Wbsite.com	02)123-4567	support@Website.com

(4) 준수 정보 & 제한 보증(Compliance Information & Limited Warranty)

- Serial 보존, 불법 등록 사용금지 등의 준수 사항 권고
- 저작권 정보 관련 작성(출처: Android, Microsoft)

1. 저작권자 소유권 정보

Copyright © 제품 소프트웨어 ® Corporation, 2015. 모든 권리는 저작권자의 소유입니다.

2. SW 허가권 관련 명시

Android는 Google Inc의 등록 상표입니다. 이 상표 사용 허가권은 Google에 있습니다.

Google TM, Google 로고, Android TM, Android 로고 및 Gmail TM은 Google Inc의 상표입니다. 이 상표의 사용 허가권은 Google에 있습니다.

Microsoft, MS, Windows, MS Office는 미국, 대한민국 또는 기타 국가에서의 Microsoft Corporation의 등록 상표 또는 상표입니다.

여기서 언급된 기타 다른 제품과 회사 이름은 각각 그 소유자의 상표일 수 있습니다.

3. 확장 플랫폼, 응용 Application

본 제품은 다양한 응용 프로그램 소프트웨어에 의해 확장될 수 있는 Linux에 기반을 둔 Android 플랫폼을 탑재하고 있습니다.

4. 통신 규격

Wi-Fi 및 Wi-Fi CERTIFIED 로고는 Wireless Fidelity Alliance에 등록된 상표입니다.

Bluetooth 및 Bluetooth 로고는 Bluetooth SIG, Inc가 소유한 상표입니다.

5. 저장 장치

MicroSD TM은 SD Card Association의 상표입니다.

6. 개발 언어

Java, JME 및 기타 Java 기반 마크는 미국 및 기타 국가에서 Sun Microsystems, Inc의 상표 또는 등록 상표입니다.

7. 연동 프로그램

연동 프로그램은 ABC, Inc의 등록 상표입니다. All rights reserved(모든 권리 보유). 연동 프로그램은 ABC, Inc의 상표 또는 등록 상표입니다. All rights reserved(모든 권리 보유).

8. 문서 효력

본 문서의 어떠한 부분도 ViewSonic Corporation의 사전 서면 허가 없이 어떠한 목적 및 수단으로든 복사, 재 생산 또는 전달될 수 없습니다.

9. 지적 소유권 정보

본 제품에 속하는 모든 기술 및 제품에 대한 권리는 관련 소유자의 재산입니다.

수행 내용 / 제품 소프트웨어 사용자 매뉴얼 작성하기

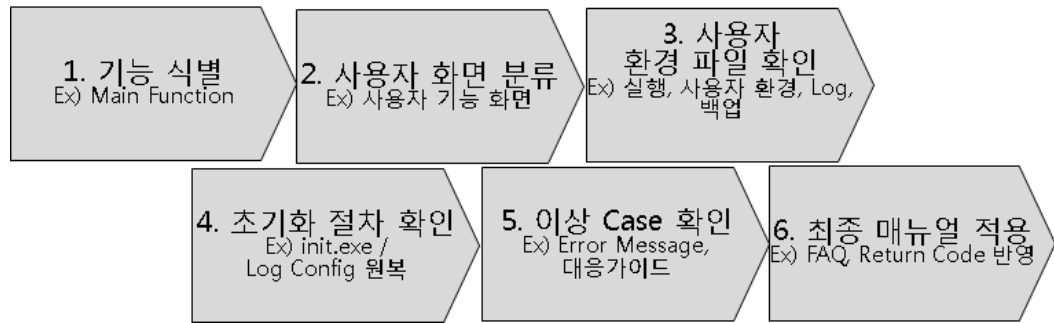
재료 · 자료

- 제품 소프트웨어 샘플
- 제품 기능 정의서, UI 정의서, 사용자 지침서
- 사용자 환경과 맞는 최소 시스템 사양

기기(장비 · 공구)

- 컴퓨터, 인터넷
- Office, Document 구동 프로그램

수행 순서



[그림 2-3] 사용자 매뉴얼 작성순서

① 제품 소프트웨어의 개요 및 사용자 기능들을 식별한다.

1. 제품 소프트웨어 개발 목적 및 기능 설명

제품 소프트웨어 자체의 목적 및 내용을 설명할 수 있도록 한다.

2. 제품 소프트웨어의 사용자 활용 기능 설명

제품의 전체적인 기능을 흐름순으로 정리하여 설명한다.

② 화면 단위로 작성될 사용자 화면을 분류한다.

1. 실제 사용자에게 보이는 화면 단위 및 메뉴 분류

화면 중심으로 UI와 메뉴를 순서대로 분류한다.

2. 사용자 메인 화면에 기초한 항목 분류

사전에 작성된 UI 정의서를 항목별로 분류한다.

3. 사용자 매뉴얼에 작성될 순서대로 UI를 분류

사용자 매뉴얼 기준의 양식을 목차에 맞게 UI를 분류한다.

③ 사용자 로그 및 백업 등 환경 관련 파일을 확인한다.

1. 로그 파일 및 백업 파일명 확인

실제로 제품을 설치할 파일(exe) 및 백업 파일명을 확인한다.

2. 로그 파일 및 백업 파일명 폴더 위치

실제로 제품을 설치할 파일(exe) 및 백업 파일명의 폴더 위치를 확인한다.

3. 파일별 기능 (실행, 사용자 환경, Log, 백업)

실행, 환경, Log, 백업 등의 다양한 파일 들을 확인하고 기능을 숙지한다.

④ 사용 초기화를 위한 절차를 확인한다.

1. 초기화 진행 시 수행 절차

제품의 제거 시를 고려하여 Uninstal 파일을 설명한다.

2. 사용자 환경 및 로그 파일 초기화

사용자 환경 및 로그 파일 초기화 단계를 순서대로 기술한다.

3. 사용자 환경과 상관없이 초기화하여 재사용을 위한 절차

설치 전 상태로의 원복 단계 절차를 기술한다.

⑤ 사용 시 이상 메시지 등에 대한 조치 사항을 확인한다.

1. 사용 시 발생하는 이상 Case Test

설치 시 발생하는 이상 현상을 발생하고 관련 테스트를 수행한다.

2. 다양한 Case명 대응 방향 정리 및 매뉴얼 작성

다양한 이상 현상 발생 시 이에 따른 대응 방향 및 매뉴얼을 정리한다.

3. Case별로 발생하는 메시지가 정상적인지 확인

다양한 이상 현상의 내용에 맞는 메시지가 간결하고 정상적으로 표시되는지 확인한다.

⑥ 사용자 대응을 위한 결과 및 코드를 캡처후 최종 매뉴얼에 적용한다.

1. 사용자 메시지 발생 시 FAQ 작성 및 대응 사항 확인

메시지에 대한 빈번한 문의 답변 및 대응 사항을 정리한다.

2. 결과에 대한 코드값 정리

결과에 대한 코드값을 취합하여 정리한다.

3. 최종 사용자 매뉴얼을 확인하고 지원 방향 상술

최종 완성된 사용자 매뉴얼을 검토하고 추가 지원 방향을 기술한다.

수행 tip

- 사용자 매뉴얼 작업 시 전체적인 기능이 사용자의 이해가 쉽도록 진행되는지가 중요하다. 가장 기본이 되어야 할 고객 편의성 측면과 이에 따른 기술적 지원 포인트를 도출해 내는 것이 중요한 포인트이다.

2-3. 제품 소프트웨어 매뉴얼의 배포용 미디어 제작

학습 목표

- 사용자가 제품 소프트웨어를 설치하고 사용하는 데 필요한 제품 소프트웨어의 설치 파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.

필요 지식 /

① 제품 소프트웨어 배포본의 개요

제품 소프트웨어 배포본은 최종 완성 단계에서 사용자가 정상 사용할 수 있도록 반드시 공식적인 인증 절차를 통하여 사용자에게 배포되도록 한다.

1. 제품 소프트웨어 배포본의 개념

제품 소프트웨어 배포본은 개발된 컴포넌트 또는 패키지에 대해 제품화하고 배포 정보를 포함하여 진행하는 것으로 버전, 시스템이 설치 및 운영을 위한 요구 사항, 설치 방법, 달라진 기능, 알려진 버그 및 대처 방법 등을 포함하여 배포한다.

2. 제품 소프트웨어 배포본의 중요 사항

- (1) 제품 소프트웨어의 배포본은 최종 완성된 제품으로 안정성을 고려하여 배포한다.
- (2) 신규 및 변경을 고려하여 배포본에는 고유 버전 및 배포 단위의 기준을 정한다.
- (3) 배포용 미디어를 제작할 때에는 저작권 및 보안에 유의하여 제작한다.
- (4) 배포본은 자체의 고유 시리얼 넘버(Serial Number)를 반드시 부착하여 복제 및 사후 지원을 고려하여 제작한다.

② 제품 소프트웨어 매뉴얼의 배포용 미디어 제작

제품 소프트웨어 배포용 미디어 제작 시 필수 항목이 포함되어 있는지를 사전에 확인하고 제작한다. 그리고 여러 미디어에 따라 특성에 맞추어 제작한다.

1. 제품 소프트웨어의 설치 파일 및 매뉴얼의 미디어 기본 구성

배포용 미디어 제작 시 다음과 같은 항목을 포함하여 제작한다.

<표 2-10> 배포용 미디어 포함 항목

미디어 포함 항목	항목 설명
버전 정보	제품 소프트웨어 패키징의 버전 정보를 포함한다.
요구 사양	시스템이 설치되고 운영되기 위한 H/W, S/W의 사양을 포함한다.
설치 방법	설치하고 운영 가능하게 하는 방법을 포함한다.
새로운 기능	이전 버전에 비해 나아진 기능을 포함한다.
알려진 오류/대처 방법	현재까지 개발된 시스템의 알려진 오류를 기술하고, 대처 방안이 있으면 함께 포함한다.
제약 사항	현재까지 개발된 시스템이 설치, 운영 상에서 어떤 제약 사항이 있다면 함께 포함한다.

2. 배포용 미디어 제작 방법

배포용 미디어는 온라인, 오프라인으로 각각 제작할 수 있으며, 각 유형별로 특성에 맞추어 제작한다.

(1) 오프라인 미디어

CD와 같은 오프라인 상으로 제품을 배포 가능하도록 제작한다.

CD에는 반드시 고유의 시리얼 넘버(Serial Number)를 포함하여 복제 등의 불법 유통을 방지한다. 시리얼 넘버(Serial Number)는 이를 체계적으로 등록 관리 해야 하며, 미디어 제작과는 별개로 이에 대한 관리 시스템을 사전에 확보해야 한다. 배포본에는 패키지 설치 기준의 Install 버전을 작성하고, Document 파일은 사용자 매뉴얼 및 설치 매뉴얼이 전부 포함될 수 있도록 한다.

<표 2-11> 배포용 미디어 제작 단계

미디어 제작 단계	항목 설명
오프라인 미디어 준비	공 CD 등 제작할 오프라인 미디어를 준비
시리얼 넘버(Serial Number) 확인	제품 고유의 시리얼 넘버(Serial Number) 등록 및 관리
매뉴얼 확인	설치 매뉴얼, 사용자 매뉴얼
Install 파일 확인	제품 설치용 Install 파일의 정상 준비 확인
제작, 정상 동작 확인	제품 CD 정상 제작 및 동작 확인

(2) 온라인 미디어

최근에는 온라인 상에도 제품의 업그레이드나 패치 버전 등을 만들어 배포한다. 업그레이드나 오류 패치, 기능 수정 등은 빈번하게 일어날 수 있으므로, 사용자의 편의를 위해 정식 버전에 한하여 제품 지원을 계속한다.

따라서 온라인 상에서 제작되는 미디어는 그 특성에 맞도록 실행 파일로 통합하여 제작한다. 또 이때에는 이전 버전 및 추가 버전에 대해서도 통합 관리를 해야 한다. 온

라인 상으로 제작되어 제공되는 내용들은 다음과 같다.

- (가) 기능 업그레이드
- (나) 제품 패치
- (다) 기능적 오류 수정
- (라) OS 등의 업그레이드에 따른 호환성 제공

수행 내용 / 제품 소프트웨어 배포용 미디어 제작하기

재료 · 자료

- 제품 소프트웨어 배포본
- 구현 명세서, 사용자 지침서
- 작성용 배포용 미디어

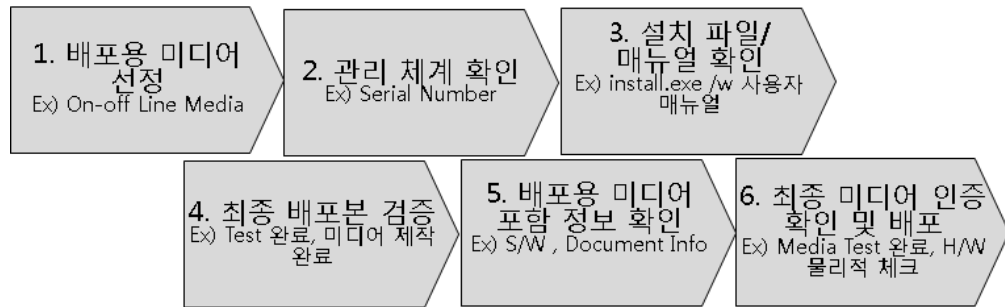
기기(장비 · 도구)

- 컴퓨터, 인터넷,
- Office, Document 구동 프로그램
- 미디어작성용 SW 및 ODD

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서



[그림 2-4] 배포용 미디어 제작순서

① 제품 소프트웨어 배포용 미디어를 선정한다.

1. 온라인, 오프라인 해당 미디어를 선정
배포용 미디어 선정을 위해 온라인, 오프라인을 전부 조사하여 선정한다.
2. 미디어 작성용 SW 및 HW를 통해 제작
미디어 작성은 SW 및 HW를 총망라하여 제작한다.

② 시리얼 넘버(Serial Number) 등록 및 관리 체계를 확인한다.

1. 시리얼 넘버(Serial Number)의 부여 체계
시리얼 넘버(Serial Number)의 체계 및 룰을 사전에 정의한다.
2. 관리를 위한 시리얼 넘버(Serial Number)의 등록
시리얼 넘버(Serial Number)를 일괄 관리할 수 있도록 등록 및 관리 체계를 수립한다.
3. 시리얼 넘버(Serial Number)의 제품 관리 시스템 사전 준비
시리얼 넘버(Serial Number) 관리 체계를 정립하고 이를 시스템화한다.

③ 설치 파일 및 매뉴얼을 최종 확인한다.

1. 설치 파일은 실제 Install이 정상 가능한 파일
설치 파일을 확인하고 Install이 정상 가능한 실행 파일인지 확인한다.
2. 설치 매뉴얼 및 사용자 매뉴얼
설치 매뉴얼 및 사용자 매뉴얼을 확인하여 미디어 제작 전 이상이 없는지 최종 체크한다.

④ 배포용 미디어로 제작되기 전의 최종 배포본이 명세서대로 구현되고 테스트되었는지를 확인한다.

1. 배포용 미디어 제작용 최종 버전은 테스트 완료된 완성 버전

배포용 미디어 제작용 버전은 테스트가 완료되었는지 여부를 최종 확인한다.

2. 미디어 제작 시 최종 확인하는 절차

미디어 제작 시 사전에 최종 버전, 테스트 완료 여부, 매뉴얼 완료 여부를 최종 확인한다.

⑤ 사용자가 배포용 미디어를 이용하여 제품을 사용하는 데 충분한 정보를 담고 있는가를 확인한다.

1. 제품 설치 및 사용에 이상이 없도록 SW 및 Document 정보 포함

배포용 미디어에서 SW 및 Document 정보를 추가 확인한다.

2. SW

설치(Install)를 위한 SW를 의미한다.

3. Document

사용 지원을 위한 매뉴얼을 의미한다.

⑥ 미디어로 제작되기 전에 인증 활동을 통하여 인증을 받았는지 확인 후 배포한다.

1. SW 인증

최종 완성본의 테스트 완료되었는지 확인한다.

2. Media 인증

미디어 작성용 CD 및 ODD에는 이상이 없는지 확인한다.

수행 tip

- 배포용 미디어 제작 작업 시 무조건 미디어를 제작하는 것이 아닌 최종 버전이 맞는지의 확인이 중요하다. 최종 버전 그리고 안정적인 버전이 배포용 미디어에 그대로 담겨지지 못하면, 미디어 자체를 다시 제작해야 할 수도 있다. 또 시리얼 넘버(Serial Number)의 등록 관리 체계를 통해 배포용 미디어의 복제 등에도 주의해야 한다.

학습 2 교수 · 학습 방법

교수 방법

- 교수자의 주도로 제품 소프트웨어의 배포본을 제공하고, 간략한 내용을 설명한다.
- 개발자가 어려워하는 설치 매뉴얼 작성을 항목 위주로 전달되도록 수업을 진행한다.
- 설치 매뉴얼 실습 시 사용자 관점에서 직접 순서대로 작성된 내용을 공유하도록 하여 설치 매뉴얼 작업 시 무엇을 중시해야 하는지 의견을 나눌 수 있도록 한다.
- 개발자가 어려워하는 사용자 매뉴얼 작성을 항목 위주로 전달되도록 수업을 진행한다.
- 사용자 매뉴얼 실습 시 사용자 관점에서 메인 화면 및 기능을 공유하도록 하여, 사용자 매뉴얼 작업 시 무엇을 중시해야 하는지 의견을 나눌 수 있도록 한다.
- 배포용 미디어 제작용 SW 및 ODD 등의 HW를 제공한다.
- 제작용 SW 등의 사용 방법을 실습하고, 제작 시 진행의 흐름에 맞게 순차적으로 진행하도록 가이드한다.
- 배포본에는 SW 및 매뉴얼이 전부 포함되었는지 확인할 수 있도록 한다.
- 배포용 미디어를 스스로 제작하고 각자 완성분을 얻는 시간이 되도록 수업을 진행한다.
- 불법 복제 등의 제작 시 유의할 사항을 느껴 보고, 사용자의 입장이 되어 미디어를 받아보기 위하여 무엇을 중시해야 하는지 의견을 나눌 수 있도록 한다.

학습 방법

- 최종 배포본을 통해 제품의 미디어 배포 과정을 알 수 있고, 절차를 이해한다.
- 설치 환경 및 고객 편의성의 중요성을 각자 정리하여 동료들과 공유한다.
- 사용자 중심의 순차적인 작업이 진행될 수 있도록, 최종 결과물에 대해 Feedback을 남길 수 있도록 실습을 진행한다.
- 사용자 화면, 환경 및 고객 편의성의 중요성을 각자 정리하여 동료들과 공유한다.

- 사용자 중심의 매뉴얼이 순차적으로 작성될 수 있도록 하고, 최종 결과물에 대해 Feedback을 남길 수 있도록 실습을 진행한다.
- 직접 배포용 미디어를 제작해 봄으로써, 제품 소프트웨어의 특징을 이해한다.
- 그간 준비했던 배포본 및 매뉴얼을 미디어로 제작함으로써 중요성을 인식한다.
- 각자 직접 배포용 미디어를 제작해 보고 소감을 정리하여 동료들과 공유한다.
- 시리얼 넘버(Serial Number) 등을 자체 부여하고 이에 대해 체계적인 제작이 진행될 수 있도록 하고, 이에 Feedback을 남길 수 있도록 실습을 진행한다.

학습 2 평 가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 설치매뉴얼 구성 수립 및 작성	- 사용자가 제품 소프트웨어를 설치하는 데 참조할 수 있도록 제품 소프트웨어 설치 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.			
제품 소프트웨어 사용자매뉴얼 구성수립 및 작성	- 사용자가 제품 소프트웨어를 설치하는 데 참조할 수 있도록 제품 소프트웨어 사용자 매뉴얼의 기본 구성을 수립하고 작성할 수 있다.			
제품 소프트웨어 매뉴얼의 배포용 미디어 제작	- 사용자가 제품 소프트웨어를 설치하고 사용하는 데 필요한 제품 소프트웨어의 설치 파일 및 매뉴얼을 배포용 미디어로 제작할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 설치매뉴얼 구성 수립 및 작성	- 제품 소프트웨어 설치 매뉴얼의 작성 항목 숙지 능력 - 제품 소프트웨어 설치 매뉴얼 항목별 가이드 제시수준 - 제품 소프트웨어 설치 순서 및 그림/표 등을 통한 사용자 중심의 매뉴얼 작성 능력			
제품 소프트웨어 사용자매뉴얼 구성수립 및 작성	- 제품 소프트웨어 사용자 매뉴얼의 작성 항목 숙지능력 - 제품 소프트웨어 사용자 매뉴얼 항목별 가이드 제시수준 - 제품 소프트웨어 사용자 관점에 맞는 그림/표 및 사용자 환경 등을 통한 사용자 중심의 매뉴얼 작성 능력			
제품 소프트웨어 매뉴얼의 배포용 미디어 제작	- 제품 소프트웨어의 최종본 분류 및 최종 SW 인증 완수 능력 - 제품 소프트웨어의 설치 파일 및 매뉴얼의 배포용 미디어 선택 및 제작 능력 - 배포용 미디어로 제작 시 저작권, DRM 등을 반영한 제작 능력			

• 구두 발표

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 설치매뉴얼 구성 수립 및 작성	<ul style="list-style-type: none"> - 제품 소프트웨어 설치 매뉴얼의 작성 항목 숙지 능력 - 제품 소프트웨어 설치 매뉴얼 항목별 가이드 제시 수준 - 제품 소프트웨어 설치 순서 및 그림/표 등을 통한 사용자 중심의 매뉴얼 작성 능력 			
제품 소프트웨어 사용자매뉴얼 구성수립 및 작성	<ul style="list-style-type: none"> - 제품 소프트웨어 사용자 매뉴얼의 작성 항목 숙지 능력 - 제품 소프트웨어 사용자 매뉴얼 항목별 가이드 제시 수준 - 제품 소프트웨어 사용자 관점에 맞는 그림/표 및 사용자 환경 등을 통한 사용자 중심의 매뉴얼 작성 능력 			
제품 소프트웨어 매뉴얼의 배포용 미디어 제작	<ul style="list-style-type: none"> - 제품 소프트웨어의 최종본 분류 및 최종 SW 인증 완수 능력 - 제품 소프트웨어의 설치 파일 및 매뉴얼의 배포용 미디어 선택 및 제작 능력 - 배포용 미디어로 제작 시 저작권, DRM 등을 반영한 제작 능력 			

피드백

1. 핵심 용어 및 학습 목표 피드백

- 제품 소프트웨어 설치 매뉴얼의 작성 항목을 숙지하여, 순서대로 작성한다.
- 제품 소프트웨어 사용자 관점에 맞는 가이드대로 사용자 중심의 매뉴얼을 작성한다.
- 제품 소프트웨어 매뉴얼은 제품이 아닌 사용자 환경을 중심으로 매뉴얼을 작성한다.
- 제품 소프트웨어의 설치 파일 및 매뉴얼은 항상 최종본으로 적용하여 배포용 미디어로 제작해 낸다.

2. 평가 항목 피드백

- 제품 소프트웨어 설치 매뉴얼의 작성 항목 숙지하여 매뉴얼 작성 능력을 향상시킨다.
- 제품 소프트웨어 설치 매뉴얼 항목별로 가이드를 제시하여 매뉴얼 수준을 높여 작성한다.
- 제품 소프트웨어 설치 순서 및 그림/표 등을 활용하여 사용자 중심의 이해도 높은 매뉴얼을 작성한다.
- 제품 소프트웨어 사용자 매뉴얼의 작성 항목을 숙지하여 매뉴얼 작성 능력을 향상시킨다.
- 제품 소프트웨어 사용자 매뉴얼 항목별로 가이드를 제시하여 매뉴얼 수준을 높여 작성한다.
- 제품 소프트웨어 사용자 관점에 맞는 그림/표 등을 활용하여 사용자 중심의 이해도 높은 매뉴얼을 작성한다.
- 제품 소프트웨어의 최종본 분류해 내고 SW 인증 개념을 반영하여 배포본을 작성해 낸다.
- 먼저 제품 소프트웨어의 설치 파일 및 매뉴얼의 배포용 미디어 선택하고, 특성에 맞게 제작하는 기법을 향상시킨다.
- 배포용 미디어로 제작 시 저작권, DRM 등을 정확히 이해하고, 이를 반영하여 제작할 수 있는 기법을 향상시킨다.

3. 구두 발표를 통한 피드백 공유

- 발표한 내용 중 미비 사항이나 보완해야 할 사항을 정리하여 돌려준다.

학습 1	제품 소프트웨어 패키징하기(LM2001020209_14v2.1)
학습 2	제품 소프트웨어 매뉴얼 작성하기(LM2001020209_14v2.2)
학습 3	제품 소프트웨어 버전 관리하기 (LM2001020209_14v2.3)

3-1. 제품 소프트웨어 버전 등록

학습 목표

- 형상 관리 지침을 활용하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다.

필요 지식 /

① 제품 소프트웨어 패키징의 형상 관리

1. 제품 소프트웨어의 형상 관리 파악

제품 소프트웨어에서 형상 관리 목적은 기본적으로 작성된 패키지의 변경 내용을 관리하기 위함이다. 이는 SW의 변화를 시간에 따라 기록하고 특정 시점의 버전을 다시 꺼낼 수 있도록 관리하는 체계를 의미한다. S/W에서 Baseline(변경 통제 시점)을 정하고, 변경을 철저히 관리 통제하는 것이 중요하며, 이에 따라 전체적인 버전 체계가 관리된다.

2. 제품 소프트웨어의 형상 관리 중요성

- (1) 제품 소프트웨어는 지속적으로 변경되는데 이에 대한 개발 통제가 중요하다.
- (2) 제품 소프트웨어의 형상 관리가 잘되지 않으면 배포판의 버그 및 수정에 대한 추적의 결여 및 무절제한 변경이 난무할 수 있다.
- (3) 형상 관리가 잘되지 않으면 제품 소프트웨어의 가시성(Visibility)의 결핍이 일어난다. 전체적인 조망이나 Insight가 결여되어 장기적인 관리 체계에 문제를 야기할 수 있다.

3. 제품 소프트웨어의 형상 관리 역할

- (1) 형상 관리를 통해 이전 리비전이나 버전에 대한 정보에 언제든지 접근 가능하여 배포본 관리에 아주 유용
- (2) 불필요한 사용자의 소스 수정 제한
- (3) 동일한 프로젝트에 대해 여러 개발자 동시 개발 가능
- (4) 에러가 발생했을 경우 빠른 시간 내에 복구 가능

(5) 사용자의 요구에 따라 적시에 최상의 소프트웨어 공급

② 제품 소프트웨어의 버전 등록 상세 기법

제품 소프트웨어 버전 관리는 SW 개발과 관련하여 코드와 라이브러리, 관련 문서 등 시간의 변화에 따른 변경을 관리하는 전체 활동을 의미한다.

1. 버전 관리 등록 관리에서 자주 사용되는 용어 정리(Terms)

버전 관리와 관련하여 여러 가지 관련 항목들을 이해하고 이에 맞는 버전 등록 및 갱신, 수정 등을 통해 버전을 관리한다.

<표 3-1> 버전 관리 항목

버전 관리 항목	항목 설명
가져오기(Import)	버전 관리지 않은 로컬 디렉토리 파일을 처음으로 저장소(Repository)에 복사
체크아웃(Check-out)	저장소(Repository) 파일 받기
체크인(Check-in)	저장소(Repository)에 새로운 버전으로 갱신
커밋 (Commit)	체크인 시 이전 갱신 사항이 있는 경우 충돌(conflict) 알림, diff 도구 이용 수정, Commit 과정 수행
저장소(Repository)	파일의 현재 버전과 변경 이력 정보를 저장하는 저장소

이러한 용어들을 기반으로 Git를 예로 들었을 때 Command 상에 다음과 같은 여러 가지 추가 명령어들을 볼 수 있다.

```
$ git
usage: git [--version] [--help] [-C <path>] [-c name=value]
       [--exec-path[=<path>]] [--html-path] [--man-path] [--info-path]
       [-p|--paginate|--no-pager] [--no-replace-objects] [--bare]
       [--git-dir=<path>] [--work-tree=<path>] [--namespace=<name>]
       <command> [<args>]

The most commonly used git commands are:
  add           Add file contents to the index
  bisect        Find by binary search the change that introduced a bug
  branch        List, create, or delete branches
  checkout      Checkout a branch or paths to the working tree
  clone         Clone a repository into a new directory
  commit        Record changes to the repository
  diff          Show changes between commits, commit and working tree, etc
  fetch         Download objects and refs from another repository
  grep          Print lines matching a pattern
  init          Create an empty Git repository or reinitialize an existing one
  log           Show commit logs
  merge         Join two or more development histories together
  mv            Move or rename a file, a directory, or a symlink
  pull          Fetch from and integrate with another repository or a local branch

  push          Update remote refs along with associated objects
  rebase        Forward-port local commits to the updated upstream head
  reset         Reset current HEAD to the specified state
  rm            Remove files from the working tree and from the index
  show          Show various types of objects
  status        Show the working tree status
  tag           Create, list, delete or verify a tag object signed with GPG

'git help -a' and 'git help -g' lists available subcommands and some
concept guides. See 'git help <command>' or 'git help <concept>'
to read about a specific subcommand or concept.
```

[그림 3-1] Git Command 상의 여러 가지 명령어

2. 작업 단계별 버전 등록 기법

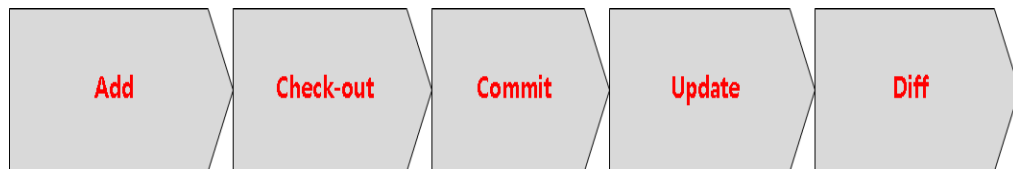
최종 인증을 패키징 버전에 한하여 컴포넌트 저장소에 등록을 한다. 또 등록된 컴포넌트는 권한을 부여 받은 사용자에게 의하여 사용될 수 있다. 무엇보다 중요한 것은 기존에 구현되고 등록되어 있는 컴포넌트의 버전 관리에 신경을 써야 한다.

- 컴포넌트 저장소(Repository)

인증 받은 컴포넌트를 등록하는 저장소로서 사용자는 저장소를 통하여 손쉽게 컴포넌트를 이용할 수 있다. 저장소는 컴포넌트의 최신 버전을 유지하고 있으며, 컴포넌트의 버전별 상태도 유지하고 관리함으로써 사용자가 컴포넌트 이용을 용이하도록 한다.

버전 등록 시 가장 중요한 점은 기 보유 버전과 새로 등록하고자 하는 버전보다 더 통합 버전 관리가 되고 있는가에 있다. 따라서 버전 등록 시에 항상 이러한 부분을 체크할 수 있도록 하고, 체크리스트를 만들어 모든 개발자가 공유할 수 있도록 한다.

일반적으로 버전 관리 도구에서 사용되는 버전 등록 순서는 다음과 같다.



[그림 3-2] 버전 등록 순서도

(1) 추가(Add)

개발자가 신규로 어떤 파일을 저장소(Repository)에 추가(Add)한다.

(2) 인출(Check-out)

추가되었던 파일을 개발자가 인출(Check-out)한다.

(3) 예치(Commit)

개발자가 인출된 파일을 수정한 다음, 저장소에 예치(Commit)하면서 설명을 붙인다.

(4) 동기화(Update)

Commit 작업 이후 새로운 개발자가 자신의 작업 공간을 동기화(Update)한다. 이때 기존 개발자가 추가했던 파일이 전달된다.

(5) 차이(Diff)

새로운 개발자가 추가된 파일의 수정 기록(Change Log)을 보면서 기존 개발자가 처음 추가한 파일과 이후 변경된 파일의 차이를 본다(Diff).

수행 내용 / 제품 소프트웨어 버전 등록하기

재료 · 자료

- 제품 소프트웨어 신규/수정본 샘플
- 버전 관리 도구
- 저장소(Repository)

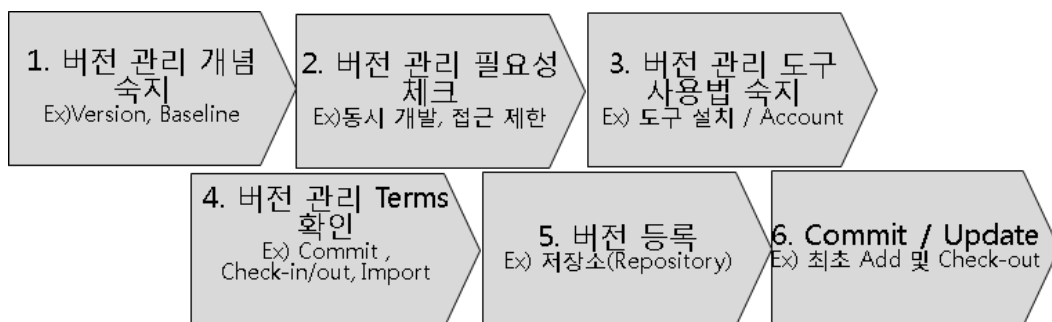
기기(장비 · 공구)

- 컴퓨터, 인터넷
- 버전 관리 구동 프로그램

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서



[그림 3-3] 버전 등록 순서

① 버전 관리의 개념 및 의미를 파악한다.

1. 버전의 의미 및 관리의 필요성

버전의 정의 및 버전 관리의 필요성을 파악한다.

2. 변경 관리 및 기준선

변경 관리 및 변경에 대한 기준선의 의미를 파악한다.

② 버전 관리가 필요한 이유를 체크한다.

1. 버전 관리를 통한 동시 개발 및 수정

버전 관리 개념을 통해 개발자가 동시에 개발 및 수정할 상황을 적용한다.

2. 정보에 대한 편리한 접근

여러 가지 개발 버전에 대해 다양한 정보를 버전별로 확인할 수 있다.

3. 권한 체크를 통한 불필요한 사용자의 접근 제한

권한 제어를 통해 허가된 개발자 이외에 접근을 제한할 수 있도록 권한 관리를 학습한다.

③ 버전 관리 도구 사용법을 숙지한다.

1. 버전 관리 도구 설치 및 계정

다양한 버전 관리 도구를 설치하고 해당 계정을 세팅한다.

2. 버전 관리 도구 메뉴별 기능 숙지

버전 관리 도구의 메뉴별 다양한 기능을 학습한다.

3. 버전 관리 도구 Check-out

버전 관리 도구의 가장 기본적 기능인 Check-out을 실습한다.

④ 버전 관리의 기본 용어를 숙지한다.

1. 2절의 기본 용어 숙지

학습 내용 3-1의 필요 지식을 참조한다.

2. 용어별 해당 기능 실습

여러 가지 용어에 대해서 다양한 기능을 실습한다.

3. 단계별 버전 등록 관리

그림 3-1의 필요 지식을 참조한다.

⑤ 버전 관리 도구를 통해 버전을 등록한다.

1. 신규버전으로 Add

버전 등록 시 신규버전을 Add 하여 최초 등록한다.

2. 추가파일 Check-out

최초 등록 이후 버전의 추가 Check-out을 실습한다.

3. 저장소(Repository)에 새로운 버전 갱신

Repository에 새로운 버전을 갱신하여 저장한다.

⑥ 버전 관리 도구를 통해 Commit 및 Update 한다.

1. 최초 Add 및 Check-out 이후 관련 실습

최초 Add 및 Check-out 기능을 우선 실습한다.

2. 개발자가 인출된 파일을 수정한 다음 Commit 실습

개발자가 추가 개발 및 수정 후 Commit 기능을 실습한다.

3. Commit 작업 이후 새로운 개발자가 자신의 작업 공간 Update

Commit 작업 이후 신규로 개발한 부분을 추가로 Update 실습한다.

수행 tip

- 버전 관리 도구 사용 시 신규 버전은 정상 등록하고 수정 버전은 Update하여 버전 관리가 정상 진행되는지가 중요하다. 무엇보다 버전 관리의 개념을 이해하고 용어를 정확히 알고 난 이후에 버전 관리 도구의 해당 메뉴별 기능이 동작되어야 한다.

3-2. 제품 소프트웨어 버전 관리 도구 사용

학습 목표

- 형상 관리 지침을 활용하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다.

필요 지식 /

① 제품 소프트웨어의 버전 관리 도구

1. 제품 소프트웨어 버전 관리 도구의 파악

제품 소프트웨어의 버전 관리 도구를 통해 현업에서는 다양한 방향으로 버전 관리를 진행해 간다. 버전 관리 도구를 이용하는 사례는 다음과 같다.

- 버전 관리를 기존의 개발 도구에 단순히 포함하는 형태에서 벗어나 ALM(Application Lifecycle Management)같이 전체 라이프 사이클을 관리하는 방향으로 진행 중
- ITIL 기반의 ITSM 도입으로 SW뿐 아니라 HW까지 전체적인 서비스 관점으로 버전 관리를 진행
- 버전 관리는 EAMS, PPM 등의 전사 IT 거버넌스의 한 부분으로 정의되어 비즈니스 연속성을 유지하기 위해 통합

2. 제품 소프트웨어 버전 관리 도구 사용 시 유의 사항

- (1) 효율적인 버전 관리 도구 사용을 위해서는 관리 도구의 다양한 기능적 활용, 지속적인 버전 관리와 버전 관리 기준 등이 필요하며, 버전 관리 도구 사용 관련, 문제 발생 시 해결 매뉴얼 사전 준비 필요
- (2) 제품 소프트웨어의 기능이 작을수록 버전 관리 정도를 그에 맞게 적절히 조정
- (3) 버전 관리 항목을 정하고 버전 관리 도구에 의한 변경 사항은 공식적인 합의에 의하여 실시
- (4) 배포 후 수정 중인 SW의 버전 관리 도구 사용은 신중하게 진행

② 버전 관리 도구의 유형별 특징과 사용 방법

제품 소프트웨어 버전 관리는 SW 개발과 관련하여 코드와 라이브러리, 관련 문서 등 시간의 변화에 따른 변경을 관리하는 전체 활동을 의미한다.

1. 버전 관리 도구의 유형별 특징

(1) 방식에 따른 버전 관리 도구 유형

<표 3-2> 버전 관리 도구 유형

버전 관리 도구 유형	버전 관리 도구 설명
공유 폴더 방식 (RCS, SCCS)	<ul style="list-style-type: none"> - 매일 개발 완료 파일은 약속된 위치의 공유 폴더에 복사 - 담당자 한 명이 매일 공유 폴더의 파일을 자기 PC로 복사하고 컴파일하여 에러 확인과 정상 동작 여부 확인 - 정상 동작일 경우 다음날 각 개발자들이 동작 여부 확인
클라이언트/서버 방식 (CVS, SVN)	<ul style="list-style-type: none"> - 중앙에 버전 관리 시스템이 항상 동작 - 개발자들의 현재 작업 내용과 이전 작업내용 추적 용이 - 서로 다른 개발자가 같은 파일을 작업했을 때 경고 출력 - Trac이나 CVS view와 같은 GUI 툴을 이용 모니터링 가능
분산 저장소 방식 (Git, Bitkeeper 등)	<ul style="list-style-type: none"> - 로컬 저장소와 원격저장소 구조 - 중앙의 저장소에서 로컬에 복사(clone)한 순간 개발자 자신만의 로컬저장소에 생성 - 개발 완료한 파일 수정 이후 로컬 저장소에 커밋한 이후 다시 원격 저장소에 반영(Push)하는 방식

(2) 구분에 따른 버전 관리 도구 소개

<표 3-3> 버전 관리 도구 구분

구분	유형	예시
저장소 구분	로컬 버전 관리 시스템	rsc
	중앙 집중형 버전 관리 시스템	CVS, SVN, Clear Case
	분산형 버전 관리 시스템	Git, Mercurial
소스 공개 유형	Open Source 툴	CVS, SVN
	상용 버전 관리 툴	PVCS, Clear Case

(3) 현업에서 쓰이는 다양한 버전 관리 도구별 특징

<표 3-4> 버전 관리 도구의 유형별 분류

버전 관리 도구 소개	특징
CVS (Concurrent Versions System)	<ul style="list-style-type: none"> - 서버와 클라이언트로 구성되어 다수의 인원이 동시에 범용적인 운영체제로 접근 가능하여 버전 관리를 가능케 한다. - Client가 이클립스에 내장되어 있다.
Subversion (SVN)	<ul style="list-style-type: none"> - GNU의 버전 관리 시스템으로 CVS의 장점은 이어받고 단점은 개선하여 2000년에 발표되었다. 사실상 업계 표준으로 사용되고 있으며 SVN으로 불리고 있다.
RCS (Revision Control System)	<ul style="list-style-type: none"> - CVS와 달리 소스 파일의 수정을 한 사람만으로 제한하여 다수의 사람이 파일의 수정을 동시에 할 수 없도록 파일을 잠금하는 방식으로 버전 컨트롤을 수행한다.
Bitkeeper	<ul style="list-style-type: none"> - SVN과 비슷한 중앙 통제 방식의 버전컨트롤 툴로서 대규모 프로젝트에서 빠른 속도를 내도록 개발되었다.
Git	<ul style="list-style-type: none"> - 기존 리눅스 커널의 버전 컨트롤을 하는 Bitkeeper를 대체하기 위해서 나온 새로운 버전 컨트롤로 현재의 리눅스는 이것을 통해 버전 컨트롤이 되고 있다. Git는 속도에 중점을 둔 분산형 버전 관리 시스템(DVCS)이며, 대형 프로젝트에서 효과적이고 실제로 유용하다. Git는 SVN과 다르게 Commit은 로컬 저장소에서 이루어지고 push라는 동작으로 원격 저장소에 반영된다.(로컬 저장소에서 작업이 이루어져 매우 빠른 응답을 받을 수 있다.) 또 받을 때도 Pull 또는 Fetch로 서버에서 변경된 내역을 받아 올 수 있다. 주요 특징은 다음과 같다. 1) Git의 작업 폴더는 모두 전체 기록과 각 기록을 추적할 수 있는 정보를 포함하는 완전한 형태의 저장소 2) 네트워크에 접근하거나 중앙 서버에 의존하지 않음. 3) Git는 GNU 일반 공중 사용 허가서 v2 하에 배포되는 자유 소프트웨어
Clear Case	<ul style="list-style-type: none"> - IBM에서 제작되었다. - 복수 서버, 복수 클라이언트 구조이며 서버가 부족할 때 필요한 서버를 하나씩 추가하여 확장성을 기할 수 있다.

2. 형상 관리 지침과 연계한 버전 관리 도구 사용 방법

(1) 버전 관리 도구 도입 사용 사례

위에서 다양한 유형의 버전 관리 도구에 대해 설명하였다. 이런 다양한 버전 관리 도구 중에서 제품에 맞는, 상황에 맞는 버전 관리 도구를 도입하는 것이 중요하다. 이를 위해 다음과 같이 버전 관리 도구 도입 사용 사례를 소개한다.

<표 3-5> 버전 관리 도구 사용 사례

사용 제시사례	버전 관리 도구 검토 항목
제품 소프트웨어 최종 배포본만 관리	<ul style="list-style-type: none"> - 최종 배포본 이후의 사용자 지원에 따른 버전 관리의 어려움 - 변경 관리, 추가 버전 등의 이유로 버전 관리 도구 검토
고객 지원팀의 경우 수작업 소요	<ul style="list-style-type: none"> - 최종 배포 이후 고객 지원 팀은 직접 문의에 대한 수작업 기술지원이 많은 부분 차지 - 수작업 및 유지보수 개선 관점에서의 버전 관리 도구 검토
예산 및 비용에 대한 부담	<ul style="list-style-type: none"> - 버전 관리 도구를 사용하지 않으면 인력 등의 투입으로 공수에 대한 비용 발생 - 비용에 맞는 버전 관리 도구에 대한 도입 효과 측정
제품 개발 및 사용상의 미숙으로 인한 어려움	<ul style="list-style-type: none"> - 버전 관리 도구의 사용 미숙으로 오히려 일정 및 개발복 잡도 증대 - 사용 법이 일반화되고 충분히 접근이 쉬운 버전 관리 도구의 사용법 전파

(2) 형상 관리 지침을 고려한 버전 관리 도구 사용

제품 소프트웨어의 개발 환경은 다양한 종류의 개발 도구들과 다양한 종류의 OS 환경을 지니고 있으며, 제품 배포를 위해 다양한 개발자들이 함께 작업함으로써 많은 개발 소스 파일들을 체계적으로 관리할 필요가 있다.

또 소프트웨어에 대한 형상 관리를 하지 않고서는 소프트웨어에 대한 품질을 확신하기 힘들다. 수작업을 통해 이러한 것들을 체계적으로 관리하는 데는 한계가 있고, 많은 시간과 수고와 노력이 필요하기 때문에 이런 문제를 해결하기 위해서 버전 관리 도구를 이용하여 관리한다. 내부 형상 관리 지침과 버전 관리 도구를 연계하여 필요성을 생각해 볼 수 있다. 다음의 사항을 참고하여 버전 관리 도구를 사용한다.

<표 3-6> 버전 관리 도구 사용 유의점

① 형상 관리 지침에 의거 버전에 대한 정보를 언제든지 접근할 수 있어야 한다.

제품 소프트웨어를 신규 개발하거나, 배포본 수정을 하거나, 버그 수정을 하거나 기타 여러 가지 이유로 인해 프로젝트 단위 접근이든 프로젝트를 구성하고 있는 파일 단위의 접근이든 간에 개발자가 원하는 때에 원하는 모습을 다시 구성할 수 있어야 하는 것은 가장 기본적인 사항이다. 따라서 버전 관리 도구를 통해 언제든지 접근가능하도록 그리고 언제든지 복구가 가능하도록 시기에 관계없이 버전 관리 도구를 사용할 준비가 되어 있어야 한다.

② 제품 소프트웨어 개발자, 배포자 이외에 불필요한 사용자가 소스를 수정할 수 없도록 해야 한다.

개발 및 배포 환경의 보안은 가장 중요하다. 버전 관리 도구를 사용 시에는 권한 문제가 있어서 더더욱 그러하다. 의도했든 아니든 버전 관리에 대한 권한은 버전 관리

도구 사용 시에 미리 정리되어 있지 않으면 사용 시에 문제를 일으킬 수 있다. 일례로 중요한 파일이 신입 사원의 손에 의해 권한 구분 없이 변경이 된다면, 버전 관리 도구 사용 자체의 의미가 없어질 수 있다. 따라서 이에 대한 권한 문제를 버전 관리 도구를 사용 시에 우선으로 고려해야 한다. 중요한 버전 관리 도구에 관련한 권한 체크 항목은 다음과 같다.

- 서버에 대한 권한
- 버전 관리 디렉토리의 권한
- 버전 파일의 권한
- 사용자 행위에 대한 권한

③ 버전 관리의 기본 개념: 동일한 프로젝트에 대해서 여러 개발자가 동시에 개발할 수 있어야 한다.

버전 관리 도구 사용 시 프로젝트 내부에서 두 사람 이상이 동시에 작업하는 경우가 많이 있다. 버전 관리 도구 사용 방법에 주의를 기울이지 않으면, 다른 개발자에 의해서 자신이 작업한 소스가 덮어 쓰여질 수 있게 된다. 이때 다른 개발자들 간에 어떻게 두 개의 모듈을 병합하겠는가도 생각해 볼 문제다. 단순히 두 개의 파일을 비교해서 병합한다고 해도 많은 수작업을 필요로 한다. 위에서 언급한 내용까지 감안한다면 특정 파일은 불필요한 개발자들의 접근을 막고, 다른 파일은 개발자가 동시에 접근을 해야 하고, 이런 파일들이 같은 디렉토리에 존재한다면, 단순히 디렉토리의 파일 권한 정도로는 해결이 안된다는 것을 쉽게 알 수 있다. 버전 관리 도구는 파일을 동시에 접근하는 데 해결책을 제시해 준다. 버전 관리 도구 사용 시 이와 같은 버전 관리 개념에 대해서 기본적으로 숙지하고 사용하는 것이 중요하다.

④ 에러 발생 시 최대한 빠른 시간 내에 복구한다.

제품 소프트웨어 개발 이후 배포가 완료된 시점에 발생하는 것이 에러나 버그에 대한 부분이다. 실제로 아주 완벽히 배포가 된 제품 패키지라고 해도 버그 없이 계속 서비스될 확률은 찾기 어렵다. 피해갈 수 없는 상황이므로 최대한 빠른 시간 내에 이를 복구하는 것이 해답이다. 우리는 1장부터 사용자 편의성을 고려한 제품 소프트웨어 패키징을 위해 노력해 왔다. 사용자는 최대한 빠른 시간 내에 복구를 원할 것이다. 버전 관리 도구의 효과적 사용을 통해 빠른 시간 내에 담당자가 버그에 대한 사실을 알 수 있도록 하고, 이전 버전 및 자료를 탐색하는데 걸리는 시간도 줄여 주게 된다. 버전 관리 도구의 필요성이다.

수행 내용 / 제품 소프트웨어 버전 관리 도구 사용하기

재료 · 자료

- 제품 소프트웨어 신규/수정본 샘플
- 버전 관리 도구
- 저장소(Repository)

기기(장비 · 공구)

- 컴퓨터, 인터넷
- 버전 관리 구동 프로그램

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서



[그림 3-4] 버전 관리도구 사용

① 버전 관리 도구 사용 시의 기능 및 유의 사항을 확인한다.

1. 버전 관리 도구의 고유한 기능 및 용어

버전 관리 도구의 고유한 기능을 메뉴별로 실습한다.

2. 버전 관리 도구 사용 시 유의 사항 숙지

버전 관리 도구 사용 시 유의해야 할 사항들에 대해 발표하고 공유한다.

② 버전 관리 도구의 유형별 기능적 동작 원리를 학습한다.

1. 버전 관리 도구의 유형 학습

버전 관리 도구의 유형에 대해, 학습한 부분을 표로 공유한다.

2. 버전 관리 도구의 특징에 맞는 동작 원리

각 버전 관리 도구들을 비교하고 이에 맞는 동작 원리를 공유한다.

③ 다양한 버전 관리 도구를 실행하여 특징을 이해한다.

1. 다양한 버전 관리 도구 정리

다양한 버전 관리 도구를 특성에 맞게 표로 정리한다.

2. 버전 관리 도구별 고유한 특징 이해

버전 관리 도구별로 고유한 특징 및 장단점을 발표한다.

3. 각 버전 관리 도구의 특징에 맞추어 적합한 도구 선정

버전 관리 도구들을 비교하고 필요에 맞는 가장 적합한 도구를 선정할 능력을 습득한다.

④ 버전 관리 도구별로 해당 기능을 동작하며 실습한다.

1. 버전 관리 도구별 공통적인 기능을 동작

버전 관리의 기본 개념 및 용어에 맞는 공통기능을 실습한다.

2. 버전 관리 도구별 고유한 기능을 개별 동작

버전 관리 도구별로 유일한 고유의 기능을 실습한다.

3. 각 버전 관리 도구를 통해 등록 및 Update 실습

버전 관리 도구를 등록하고 수정한 Update 기능을 실습한다.

⑤ 형상 관리의 개념 및 이에 맞는 버전 관리 도구 도입을 공유한다.

1. 형상 관리의 개념 및 지침 이해

형상 관리의 개념 및 정책을 발표하고 공유한다.

2. 프로젝트의 특징에 맞는 버전 관리 도구 사용

해당 개발프로젝트의 특징에 맞는 버전 관리 도구를 선택할 수 있도록 학습한다.

3. 버전 관리 도구의 도입을 위한 중요 사항 고려

버전 관리 도구의 선택 시 고려 사항을 발표하고 공유한다.

⑥ 버전 관리 도구 사용 시 반드시 준수해야 할 항목을 발표한다.

1. 버전 관리 도구 사용 시 필수조건 체크

버전 관리 도구 사용 시 반드시 확인해야 할 사항을 확인한다.

2. 형상 관리 정책에 맞는 버전 관리 도구 체크

형상 관리 정책을 미리 확인하고 버전 관리 도구를 결정한다.

3. 형상 관리 지침에 맞는 버전 관리 도구의 필요성 숙지

형상 관리 지침을 사전에 숙지하고 이에 맞는 버전 관리 도구의 필요성을 발표하고, 공유한다.

수행 tip

- 버전 관리 도구 사용 시 다양한 버전 관리 도구 중에서 필요한 기능을 사용할 줄 아는 것이 중요하다. 여러 가지 버전 관리 도구가 있지만 내부 형상 관리 지침에 맞는 도구를 도입하는 것이 중요한 포인트이다.

3-3. 제품 소프트웨어 버전 관리 도구 활용

학습 목표

- 버전 관리 도구를 활용하여 제품 소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.

필요 지식 /

① 제품 소프트웨어 버전 관리 도구의 활용방향

제품 소프트웨어 개발 조직이나 기 배포된 제품이 있는 조직에서는 기 보유 제품 버전과 새로 개발된 버전을 동시에 유지, 관리하여야 한다. 따라서 활용성 중심으로 버전 관리 도구의 상세 기능을 활용하여 새 버전을 유지하면서 기존 버전의 관리도 함께 수행하여야 한다. 따라서 이를 위해 백업 및 장애 복구 등의 추가적인 사항도 Case별로 고려해야 한다.

② 버전 현황 관리 기법 및 백업, 장애 복구

제품 소프트웨어 버전 관리는 SW 개발 관련하여 코드와 라이브러리, 관련 문서 등 시간의 변화에 따른 변경을 관리하는 전체 활동을 의미한다.

1. 버전 관리 도구를 활용한 통합 버전 현황 관리

버전 관리 도구를 활용한 버전 현황 관리 작업에는 기존의 고유한 버전 관리 외에도 버전 관리 도구에서 지원하는 모든 관리 작업 전체를 관리할 수 있도록 해야 한다. 여기에는 기존에 버전 관리의 버전별 공동 작업 이외에도 백업 및 복구 작업, 공동 솔루션 기반의 여러 가지 작업 등이 통합되어 현황 관리된다.

<표 3-6> 버전 현황 관리

관리 작업	활용사례	비고
버전 관리 백업 및 복구	이전 버전으로의 복구 필요시	백업, 복구
	이전 버전의 기능을 재활용 시	
동일 버전 공동 작업	동일 버전의 소스 코드를 두 명 이상의 개발자가 수정할 경우	공동 작업
	분산환경에서 두 명 이상의 개발자가 동일한 버전의 소스를 수정할 경우	분산 버전 관리 툴 필요
여러 버전 솔루션 작업	하나의 솔루션을 기반으로 여러 버전의 제품을 개발 할 경우	브랜치 관리 기능 필요

2. 버전 관리 도구 활용한 제품 소프트웨어 소스 및 자료 백업

(1) 제품 소프트웨어 소스 및 자료 백업의 의미

소스 및 자료 백업의 목적은 크게는 천재지변이나 해킹, 작게는 소프트웨어 버그 및 오동작 등으로 인한 각종 재해에 대비해 System을 Recovery 가능한 상태로 만들기 위해 준비하는 것이다.

손실의 예방 및 최소화를 우선해야 하고, 정작 재해가 발생했을 시는 최단 기간에 정상 업무에 복구하도록 버전 관리 도구를 적절히 사용할 수 있어야 한다. 또 버전 관리 도구의 사용 방법을 숙지하여 복구 과정의 효과적 통제/관리가 이루어질 수 있도록 해야 한다. 이를 통해 제품 소프트웨어의 소스 및 자료를 정보 자산으로 훌륭하게 보호할 수 있다.

(2) 버전 관리 도구 활용 시 자료 백업 정책 및 방법

버전 관리 도구를 활용, 제품 소프트웨어 백업은 내부의 정확한 백업 정책을 먼저 세워야 하고, 상세 백업 방법을 공유해야 한다. 백업 정책 사례 및 방법은 다음과 같다.

<표 3-7> 자료 백업 정책 및 방법

버전 관리 항목	항목 설명
백업 정책	<ul style="list-style-type: none">- 버전 관리 라이브러리에 대한 백업 파일은 버전 관리 라이브러리가 저장된 Disk와 분리된 Disk에 저장- Disk 백업은 1일 1회(ex. 매일 03시) 실시(단, 일요일은 제외함.)- CD 백업은 1주일 1회(ex. 매주 금요일) 실시- 실수에 의한 삭제를 예방하기 위해 백업은 최소 D-2일분 이상 보관
백업 방법	<ul style="list-style-type: none">- 변경된 부분만 백업하는 경우(증분 백업) 편리해 보이지만, 복구 시 복잡한 과정을 거쳐야 하고, 증분 백업 파일에 문제가 발생하는 경우 어려움이 생길 수 있으므로 버전 라이브러리의 백업은 빠른 복구를 위해 Full 백업- 백업 작업에 대한 임시/신규 요청 또는 중지 요청은 절차에 따라 실시- 백업결과를 버전 관리를 담당하는 CMO(Change Mgt Officer)가 주기적으로 점검하며, 그 결과를 버전 관리 정기 보고서 함께 보고

(3) 버전 관리 도구를 활용한 소스 및 자료 백업

(가) 디렉토리를 신규로 만들며 물리적 복사: Repository Backup

버전 관리 도구 Git를 사례로 들면,

Git 자체가 소스 코드 백업용으로 쓰이지만, 2차 백업 개념으로 ./.Git 폴더의 내용 디렉토리를 물리적으로 다른 곳에 보관하여 백업한다.

새로 만든 디렉토리나 이미 파일이 있는 디렉토리에서 Git init 명령을 실행하면 Git는 데이터를 저장하고 관리하는 .Git 디렉토리를 만든다. 이 디렉토리를 복사하기만 해도 저장소가 백업된다. 이처럼 1차로 가장 기본적인 원시 백업 방법으로 디렉토리 자체를 복사하여 백업한다.

(나) 버전 관리 도구의 백업 기능 활용: Git bundle

Git Repository의 bundle을 생성하는 process를 통해 백업 자체를 버전 관리 도구에
서 자동화하여 백업을 진행해 주는 백업 방법이다. 다음과 같은 방식을 갖는다.

`Git backup [-d DESTINATION DIR] [-f BUNDLENAME]`

환경값은 Git config file에 세팅을 하며, Git Repository의 bundle을 생성할 수 있게
된다. \$HOME/.Gitconfig 파일에는 Repository에 대한 구성 설정이 들어 있다. 또 백
업 이후 이를 복구할 때는 Git-restore를 활용하여 복구한다.

Git config 관련한 설정값은 다음과 같다.

```
backup.directory = string
- the directory for the backup bundles
backup.prefix-date = boolean
-- date in the format: "YYYY-MM-DD - " -- default = true
backup.prefix-time = boolean
- time in the format: "HH:MM:SS - " -- default = false
```

수행 내용 / 제품 소프트웨어 버전 관리 도구 활용하기

재료 · 자료

- 제품 소프트웨어 신규/수정본 샘플
- 버전 관리 도구
- 저장소(Repository)

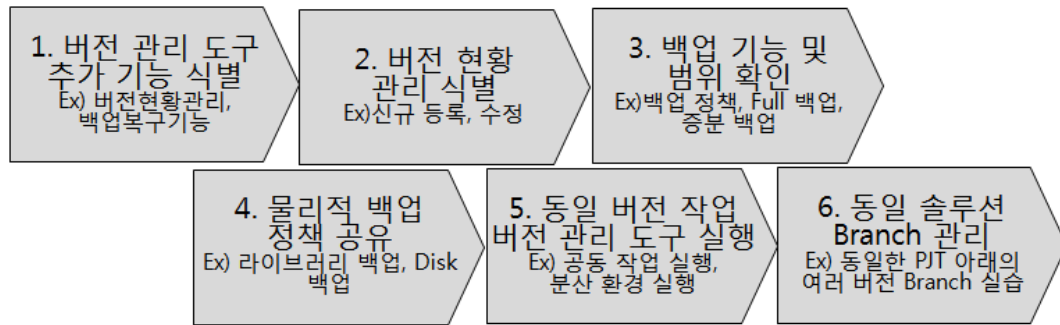
기기(장비 · 공구)

- 컴퓨터, 인터넷
- 버전 관리 구동 프로그램

안전 · 유의 사항

- 실습 후에는 컴퓨터의 전원을 끈다.

수행 순서



[그림 3-5] 버전 관리 도구 활용

① 버전 관리 도구를 통해 추가 활용할 수 있는 기능을 식별한다.

1. 버전 관리 도구의 추가 기능 확인

버전 관리 도구의 기본 기능 이외의 기능을 추가로 실습한다.

2. 버전 관리 도구의 관리 및 백업기능 식별

버전 관리 도구의 관리 및 백업 기능에 대해 학습한다.

② 버전 관리 도구를 통해 나타날 수 있는 버전 현황 관리 항목을 식별한다.

1. 기본적인 버전 현황 (신규 등록, 수정)

신규등록, 수정 등의 기본적인 버전 관리를 실습한다.

2. 동일 버전 공동 작업

동일 버전을 여러 개발자가 동시에 작업할 경우의 버전 관리를 실습한다.

3. 여러 버전 하나의 솔루션

여러 가지 버전을 하나의 도구 및 솔루션을 통해 관리하는 상황을 실습한다.

③ 버전 관리 도구를 통해 활용 가능한 백업의 기능 및 범위를 확인한다.

1. 정확한 백업 정책 수립

전체적인 백업 관리 및 정책에 대해 사전에 수립하도록 실습한다.

2. 버전 관리 도구를 통한 백업 방법 숙지

버전 관리 도구의 백업 기능을 실습한다.

3. Full 백업, 증분 백업

백업의 2가지 방법인 Full 백업, 증분 백업을 실습한다.

④ 버전 관리 도구 이외에도 물리적 백업, 백업 정책 같은 부분을 작성하여 공유한다.

1. 라이브러리 백업, Disk 백업

기본적인 백업의 기능인 라이브러리 백업, Disk 백업에 대해 학습한다.

2. 백업 주기

백업 주기에 대해 여러 가지 기준을 세워 공유한다.

3. 백업, 버전 관리 정기 보고서

백업, 버전 관리에 대해 정기적으로 수행하고, 이를 보고하는 기본 보고서를 작성해 본다.

⑤ 동일버전을 공동작업할 때 버전 관리 도구 사용 사례를 발표한다.

1. 공동 작업

동일버전의 소스를 공동 작업으로 실습한다.

2. 분산 환경 작업

분산환경에서의 동일 버전 소스를 실습한다.

3. 분산 환경에서 버전 관리 도구 사용 사례 도출

분산환경에서 버전 관리 도구의 사례를 도출하고 발표한다.

⑥ 다수의 개발자가 동일 솔루션 환경하에서 브랜치(Branch)를 관리할 수 있도록 실습해 본다.

1. 하나의 솔루션 기반 하에 여러 버전 제품 소프트웨어

하나의 솔루션에 여러 버전을 가진 상황을 실습한다.

2. Branch 관리 기능에 대한 실습

여러 가지 Branch를 실습하고 관리 기능을 학습한다.

3. 버전 관리 도구를 활용한 응용 기능에 대한 다양한 Case 정리

버전 관리 도구의 응용 기능에 대하여 정리하고 발표한다.

수행 tip

- 버전 관리 도구의 활용 시 기본 기능 이외에 도구에서 제공하는 여러 가지 추가 기능을 활용해 본다. 버전 관리 도구를 통한 백업 관리 및 이를 통한 복구에 대해서 다양한 상황을 고려하여 실습한다.

학습 3 교수 · 학습 방법

교수 방법

- 교수자의 주도로 샘플 버전의 신규 및 수정 버전을 제공하고, 간략한 내용을 설명한다.
- 버전 관리의 버전의 개념과 관리의 필요성을 설명한다.
- 버전 관리에서 쓰이는 고유한 용어들을 설명하고, 의미에 맞게 버전 관리 실습을 유도한다.
- 버전 관리 도구의 추가 기능에 대한 사용법을 설명한다.
- 버전 관리 도구에서 각 메뉴별 기능을 설명하고, 동작에 맞게 버전 관리 도구를 통해 실습을 유도한다.
- 버전 관리 도구의 다양한 유형별로 버전 등록 및 Update를 실제로 실습을 해 보고, 어떠한 특징을 가지고 있는지 의견을 나눌 수 있도록 한다.
- 버전 관리 도구에서 버전 현황 및 백업 등의 기능을 설명하고, 동작에 맞게 버전 관리 도구를 통해 실습을 유도한다.
- 버전 관리 도구의 활용 기능을 사례로 주어 실습하고, 각 실습자별 버전에 맞게 실습하도록 가이드 한다.
- 버전 관리 도구의 동일버전, 여러 가지 버전 등으로 나누어 실습을 해 보고, 어떠한 특징을 가지고 있는지 의견을 나눌 수 있도록 한다.

학습 방법

- 버전 관리 및 버전의 개념과 역할을 이해한다.
- 프로젝트에서 버전 관리를 반드시 해야 하는 이유를 이해한다.
- 버전 관리 도구에 대한 개념과 사용법을 이해한다.
- 프로젝트에서 버전 관리 도구의 기능을 사용하여 버전 관리를 직접 실습해 본다.
- 다양한 유형별로 버전 관리 도구의 사용법을 정확히 숙지한다.

- 형상 관리 지침 및 정책에 맞는 버전 관리 도구의 사용법을 공유한다.
- 각 유형별 다양한 버전 관리 도구를 사용해 보고 어떠한 Case에서 사용할 때 유리한지 느낌을 공유한다.
- 버전 관리 도구에 대한 추가 기능의 활용 사용법을 이해한다.
- 프로젝트에서 버전 관리 도구의 추가 기능을 사용하여 버전 현황 관리, 백업 관리를 직접 실습해 본다.
- 다양한 Case별로 버전 관리 도구의 활용법을 정확히 숙지한다.
- 백업 정책에 맞는 버전 관리 도구의 백업 방법을 공유한다.
- 각 사례에 맞는 버전 관리 도구 기능을 사용해 보고, 어떠한 Case에서 사용할 때 유리한지 느낌을 공유한다.

학습 3 평가

평가 준거

- 평가자는 학습자가 수행 준거 및 평가 시 고려 사항에 제시되어 있는 내용을 성공적으로 수행하였는지를 평가해야 한다.
- 평가자는 다음 사항을 평가한다.

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 버전 등록	- 형상 관리 지침을 활용하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록할 수 있다.			
제품 소프트웨어 버전 관리 도구 사용	- 형상 관리 지침을 활용하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전 관리 도구를 사용할 수 있다.			
제품 소프트웨어 버전 관리 도구 활용	- 버전 관리 도구를 활용하여 제품 소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 수행할 수 있다.			

평가 방법

- 평가자 체크리스트

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 버전 등록	- 버전 및 버전 관리에 대한 기본 개념 - 형상 관리의 개념 및 지침에 대한 숙지 능력 - 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전의 등록 수준			
제품 소프트웨어 버전 관리 도구 사용	- 버전 관리 도구의 특성 및 사용법 숙지 능력 - 다양한 버전 관리 도구의 특성별 이해 수준 - 형상 관리 지침과 연계한 상황에 맞는 버전 관리 도구 사용 능력			
제품 소프트웨어 버전 관리 도구 활용	- 버전 관리 도구를 활용한 버전 현황 관리 수행 능력 - 버전 관리 도구를 활용한 기능별 관련 자료 백업 수행 능력 - 실제 버전 관리 도구를 활용한 자료 복구 수준 평가			

• 구두 발표

학습 내용	평가 항목	성취수준		
		상	중	하
제품 소프트웨어 버전 등록	<ul style="list-style-type: none"> - 버전 및 버전 관리에 대한 기본 개념 - 형상 관리의 개념 및 지침에 대한 숙지 능력 - 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전의 등록 수준 			
제품 소프트웨어 버전 관리 도구 사용	<ul style="list-style-type: none"> - 버전 관리 도구의 특성 및 사용법 숙지 능력 - 다양한 버전 관리 도구의 특성별 이해 수준 - 형상 관리 지침과 연계한 상황에 맞는 버전 관리 도구 사용 능력 			
제품 소프트웨어 버전 관리 도구 활용	<ul style="list-style-type: none"> - 버전 관리 도구를 활용한 버전 현황 관리 수행 능력 - 버전 관리 도구를 활용한 기능별 관련 자료 백업 수행 능력 - 실제 버전 관리 도구를 활용한 자료 복구 수준 평가 			

피드백

1. 핵심 용어 및 학습 목표 피드백

- 버전 관리의 개념을 정확히 숙지하고, 그에 맞는 기능으로 제품 소프트웨어에 효과적으로 적용한다.
- 형상 관리 지침을 사전에 검토하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전을 등록한다.
- 버전 관리 도구의 유형별 특성에 맞게 활용하고, 이에 대한 기본적인 기능을 최대한 활용한다.
- 버전 관리 도구의 추가적인 기능을 활용하여 제품 소프트웨어에 대한 버전 현황 관리와 소스, 관련 자료 백업을 부가적으로 수행한다.

2. 평가 항목 피드백

- 버전 및 버전 관리에 대한 기본 개념을 확실히 숙지하여 활용능력을 향상시킨다.
- 형상 관리의 개념 및 지침에 대하여 미리 숙지하여 제품버전의 활용능력을 숙달시킨다.
- 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 버전의 등록을 직접 실습하여 신규 버전의 개념 수준을 높인다.
- 버전 관리 도구의 특성을 이해하여 도구 자체의 기능에 맞는 사용법 숙지 능력을 높인다.
- 다양한 버전 관리 도구의 유형별 특성을 이해하여 상황에 맞는 사용능력을 향상시킨다.
- 형상 관리 지침을 명확히 이해하여, 이에 연계한 버전 관리 도구의 적용 방향을 숙달시킨다.
- 버전 관리 도구의 추가 기능을 숙지하여 버전 현황 관리의 체계적 활용도를 높인다.
- 버전 관리 도구의 추가 기능을 숙지하여 백업정책 및 백업의 기법 수준을 향상시킨다.
- 사전에 백업 정책을 명확히 숙지하여 장애 시 버전 관리 도구를 통한 자료 복구 수행능력을 높인다.

3. 구두 발표를 통한 피드백 공유

- 발표한 내용 중 미비 사항이나 보완해야 할 사항을 정리하여 돌려준다.



- Allan Kelly(2008). 『Changing Software Development』 . Wiley
- Chacon, Scott, Straub, Ben(2014). 『Pro Git』 . Apress
- Duvall, Paul(2007). 『Continuous Integration』 . Addison-Wesley
- David Thomas, Andrew Hunt(2004). 『PRAGMATICVERSION CONTROLUSING CVS』 . O'REILLY
- Donald J. Reifer(2011). 『software change management : Case studies and practical advice』 . Microsoft Press
- GUNDERLOY(2004). 『CODER TO DEVELOPER』 . SYBEX
- Hundhausen, Richard(2005). 『Working With Microsoft Visual Studio 2005 Team System』 . Microsoft
- Jon Loeliger(2009). 『Version Control with Git: Powerful Tools and Techniques for Collaborative Software Development』 . O'ReillyMedia
- Michael E. Bays, Michael Bays(1999). 『Software Release Methodology』 . PrenticeHall
- Minna(2000). 『Software change management processes in the development of embedded software』 . Technical Research Centre of Finland
- Neil Matthew(2003). 『BEGINNING LINUX PROGRAMMING』 . Wiley Publishing
- Roger S. Pressman(2009). 『Software Engineering: A Practitioner's Approach』 . McGraw-Hill
- Second International Conference Sera(2004). 『Software Engineering Research And Applications : 29. MARMI-RE : A METHOD AND TOOLS FOR LEGACY SYSTEM MODERNIZATION』
- Online, <http://www.appfusions.com/display/StashSCMImporter/CVCS+vs.+DVCS+In+a+Nutshell>
- Online, <http://Git-scm.com/book/ko>



작업 포트폴리오

평가 일자 :		사번:		이름:	
1. 제품 소프트웨어 패키징		<p>제품 소프트웨어 패키징의 개념, 절차 및 릴리즈 노트에 대해 기술하고, 암호화/보안을 고려한 패키징을 설명하시오.</p> <ul style="list-style-type: none"> • • • • • 			
2. 제품 소프트웨어 매뉴얼 작성		<p>제품 소프트웨어 매뉴얼의 필요성 및 설치 매뉴얼, 사용자 매뉴얼의 구성을 기술하고, 배포용 미디어 제작 방법에 대해 설명하시오.</p> <ul style="list-style-type: none"> • • • • • 			
3. 제품 소프트웨어 버전관리		<p>제품 소프트웨어 버전의 개념 및 버전 관리의 용어 및 중요성을 기술하고, 버전 관리 도구를 유형별, 기능별로 상세히 사용법을 설명하시오.</p> <ul style="list-style-type: none"> • • • • • 			

교수자 확인: _____

NCS 학습모듈 개발진

(대표집필자)

강석진(이비스툼)

(집필진)

김보운(이화여자대학교)

김홍진(LG CNS)

유은희

장현섭((주)커리텍)

주선태(T3Q)

진권기(이비스툼)

최재준

(검토진)

김승현(경희대학교)

엄기영(우리에프아이에스)

장온순(한국IT컨설팅)

조상욱(세종대학교)

조성호(삼성카드)

(개발기관)

최가원(한국소프트웨어기술진흥협회)

이두현(한국소프트웨어기술진흥협회)

(연구기관)

옥준필(한국직업능력개발원)

김상진(한국직업능력개발원)

김성남(한국직업능력개발원)

김지영(한국직업능력개발원)

문한나(한국직업능력개발원)

홍서희(한국직업능력개발원)

*표시는 NCS 개발진임

※ 본 학습모듈은 자격기본법 시행령 제8조 국가직무능력표준의 활용에 의거하여 개발하였으며
저작권법 25조에 따라 관리됩니다.

※ 본 학습모듈은 <http://www.ncs.go.kr>에서 확인 및 다운로드할 수 있습니다.



www.ncs.go.kr