

정보처리기사 필기

소프트웨어 개발 6 제품소프트웨어패키징②

양문자 선생님

출처 : ncs 학습모듈(NCS능력단위 제품소프트웨어패키징)

소프트웨어 개발

차례

1 데이터 입출력 구현

2 통합 구현

3 제품소프트웨어 패키징

1) 제품소프트웨어 패키징

2) 제품소프트웨어 매뉴얼 작성

3) 제품소프트웨어 버전관리

4 애플리케이션 테스트 관리

5 인터페이스 구현

제품 소프트웨어 매뉴얼 작성

제품 소프트웨어의 매뉴얼

- 제품 소프트웨어 개발단계부터 적용한 기준이나 패키징이후 설치 및 사용자 측면의 주요 내용 등을 문서로 기록한 것
- 사용자 중심의 기능 및 방법을 나타낸 설명서와 안내서

설치 매뉴얼 작성의 기본 사항

- (1) 설치 매뉴얼은 개발자의 기준이 아닌 **사용자**의 기준으로 작성한다.
- (2) 최초 설치 실행부터 완료까지 **순차적**으로 진행한다.
- (3) 각 단계별 메시지 및 해당 화면을 순서대로 전부 캡처하여 설명한다.
- (4) 설치 중간에 **이상 발생 시** 해당 메시지 및 에러에 대한 내용을 **분류**하여 설명한다.

문제풀이

- 소프트웨어 설치 매뉴얼에 대한 설명으로 틀린 것은?
 - ① 설치과정에서 표시될 수 있는 예외상황에 관련 내용을 별도로 구분하여 설명한다.
 - ② 설치 시작부터 완료할 때까지의 전 과정을 빠짐없이 순서대로 설명한다.
 - ③ 설치 매뉴얼은 개발자 기준으로 작성한다.
 - ④ 설치 매뉴얼에는 목차, 개요, 기본사항 등이 기본적으로 포함되어야 한다.

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

제품 소프트웨어 설치 매뉴얼의 기본 작성 항목

기본 작성 항목		설 명																		
목차 및 개요	목차	매뉴얼 전체의 내용을 순서대로 요약																		
	개요	- 설치 매뉴얼의 주요 특징에 대해 정리 - 설치 매뉴얼에서의 구성과 설치 방법, 순서 등에 대해 기술함																		
서문	문서 이력 정보	<table><tr><td>버전</td><td>작성자</td><td>작성일</td><td>검토자</td><td>일시</td><td>검수인</td></tr><tr><td>v0.1</td><td>홍길동</td><td>2020-10-05</td><td></td><td></td><td></td></tr><tr><td colspan="2">변경 내용</td><td colspan="4">최초 작성</td></tr></table>	버전	작성자	작성일	검토자	일시	검수인	v0.1	홍길동	2020-10-05				변경 내용		최초 작성			
		버전	작성자	작성일	검토자	일시	검수인													
		v0.1	홍길동	2020-10-05																
	변경 내용		최초 작성																	
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																			
	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 - 설치 위치																		

소프트웨어 개발

제품 소프트웨어 설치 환경 체크 항목

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

소프트웨어 개발

제품 소프트웨어 설치 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
설치 관련 파일	<ul style="list-style-type: none">- 제품 소프트웨어를 설치하기 위한 관련 파일 설명- 설치 구동을 위한 exe 실행- ini나 log 파일 같은 관련 파일
설치 아이콘	Windows 구동용 설치 아이콘 설명
프로그램 삭제	해당 소프트웨어 삭제 시 원래대로 삭제하는 방법을 설명
관련 추가 정보	<ul style="list-style-type: none">- 제품 소프트웨어 이외의 관련 설치 프로그램 정보(Build)- 관련 프로그램 제작사 추가 정보 기술

소프트웨어 개발

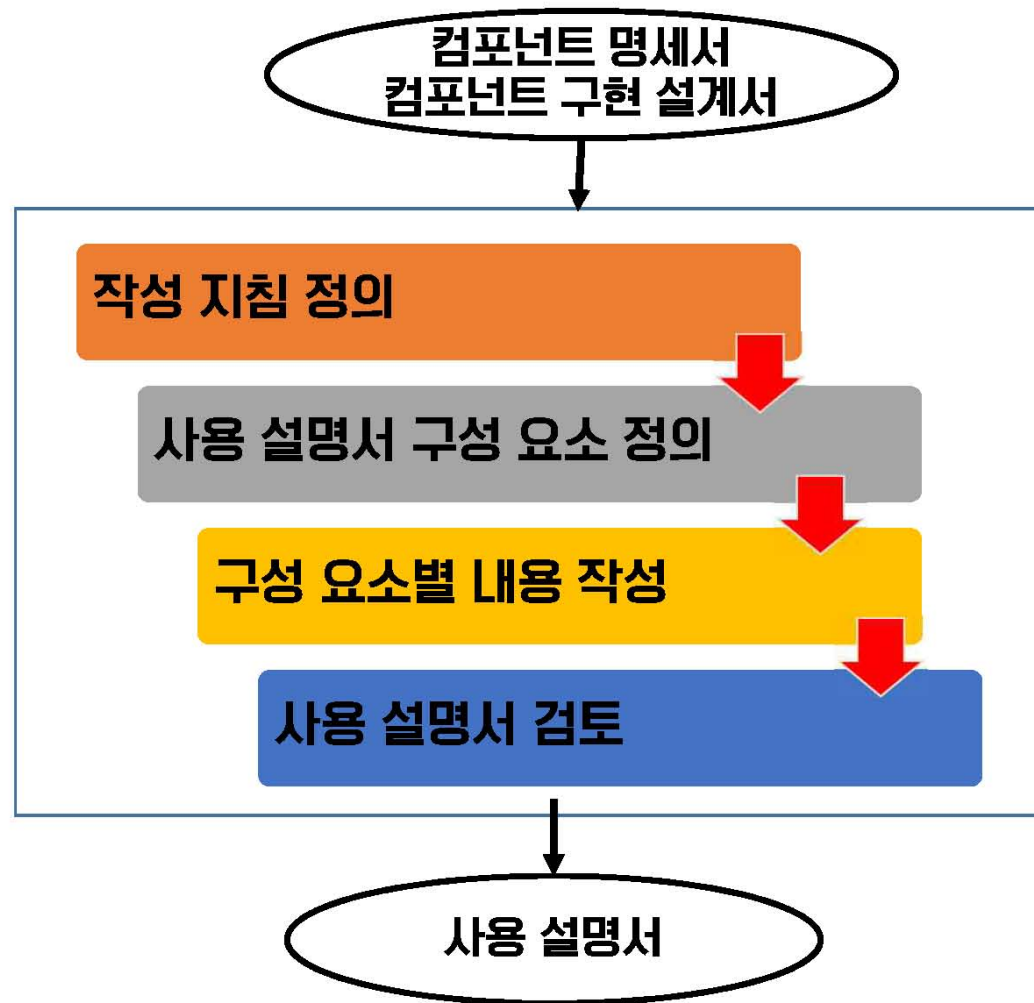
[제품 소프트웨어 설치 매뉴얼 작성 방법 상세 가이드]

기본 사항		항목 설명
설치 매뉴얼 내용	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
	설치 이상 시 메시지 설명	- 설치가 잘못되었거나 잘못된 환경일 경우 메시지 - 설치 단계별 참고 사항, 주의 사항 등을 메모로 추가
	설치 완료 및 결과	- 정상 설치 완료 시 최종 메시지 출력 - 정상 결과 및 사용에 대한 안내를 함으로써 설치가 잘되었음을 최종 통지
	FAQ	- 제품이 설치되지 않을 때를 대비
	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.
	Network 환경 및 보안	- 설치 시 Network 문제로 인해 오류가 발생하지 않도록 사전에 Connection을 체크한다. - 보안 및 방화벽 등의 문제로 설치가 안될 경우에 대비해 환경을 체크할 수 있도록 가이드한다.

소프트웨어 개발

- 고객 지원 방법(Customer Support)
 - 유선 및 E-mail, Website URL
- 준수 정보 & 제한 보증(Compliance Information & Limited Warranty)
 - Serial 보존, 불법 등록 사용 금지 등의 준수 사항 권고
 - 저작권 정보 관련 작성(출처: Android, Microsoft)

사용자 매뉴얼 작성 절차



국제 표준 제품 품질 특성

국제 표준 제품 품질 특성이란?

- 프로젝트를 진행하거나 완성된 제품에 대해 품질을 평가하는 국제 표준화 기구에서 정한 표준에 따른 기준

소프트웨어 품질의 종류

- 소프트웨어 제품품질 : 품질 자체
- 소프트웨어 프로세스 품질 : 개발 과정
- 소프트웨어 품질경영 : 조직을 지휘 관리하는 활동

소프트웨어 개발

주요 국제 표준화 기구

ISO(국제표준화기구)

설립: 1947년, 비정부기구(NGO)

역할: 지적, 과학, 기술, 경제 등 일반 분야의 국제표준 제정, 보급

IEC(국제전기기술위원회)

설립: 1906년, 비정부기구(NGO)

역할: 전기전자 분야의 국제표준 제정, 보급

ITU(국제전기통신연합)

설립: 1865년, UN 산하 전문기구

역할: 유무선 통신, 전파, 방송, 위성주파수 등에 대한 기술기준 및 표준의 개발, 보급과 국제협력 수행

https://www.ksa.or.kr/ksa_kr/942/subview.do

소프트웨어 개발

국제표준 : 소프트웨어 품질 관점

품질 표준/인증	설명
ISO/IEC 9126	품질 특성과 측정 기준 제시 (기능성/신뢰성/사용성/효율성/유지보수성/이식성) 평가방법이 없음
ISO/IEC 14598	소프트웨어 제품 평가 (표준개요/측정계획관리/개발자/획득자/평가자/평가 모델)
ISO/IEC 12119	정보기술, 패키지 소프트웨어 대한 품질 요구사항 및 시험을 위한 국제 표준 (제품 설명서, 사용자 문서, 실행 프로그램)
ISO/IEC 25000	소프트웨어 품질평가 모델인 ISO/IEC 9126와 소프트웨어 평가절차 모델 ISO/IEC 14598, ISO/IEC 12119 을 통합
Good S/W	정보통신기술협회 산하 S/W 시험 센터 주관

문제풀이

- 패키지 소프트웨어의 일반적인 제품 품질 요구사항 및 테스트를 위한 국제 표준은?

- ① ISO/IEC 2196
- ② IEEE 19554
- ③ ISO/IEC 12119
- ④ ISO/IEC 14959

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)

소프트웨어 개발

소프트웨어 품질 특성(ISO/IEC 9126)

품질특성	설명
기능성	명시된 요구와 내재된 요구를 만족하는 기능 (하위특성:적합성, 상호운용성, 보안성, 정확성, 준수성)
신뢰성	규정된 성능수준 유지 및 오류방지 (하위특성:고장허용성, 회복성, 성숙도, 준수성)
사용성	사용자에 의해 이해, 학습, 선호(하위특성:학습성, 운영성, 친밀성, 준수성)
효율성	주어진 자원을 사용하여 요구된 성능 제공(하위특성:반응시간, 자원특성, 준수성)
유지보수성	운영 환경 및 요구사항, 기능적 사양에 따른 소프트웨어 수정, 개선 (하위특성:분석성, 변경성, 안정성, 테스트 용이성, 준수성)
이식성	다양한 환경에서 운영 가능(하위특성:적응성, 설치성, 공존성, 대체성, 준수성)

문제풀이

- 소프트웨어 품질 목표 중 주어진 시간동안 주어진 기능을 오류 없이 수행하는 정도를 나타내는 것은?

- ① 직관성
- ③ 신뢰성

- ② 사용 용이성
- ④ 이식성

(2020년 3회 정보처리기사 필기 기출문제 소프트웨어 개발)

문제풀이

- ISO/IEC 9126의 소프트웨어 품질 특성 중 기능성(Functionality)의 하위 특성으로 옳지 않은 것은?

① 학습성

② 적합성

③ 정확성

④ 보안성

(2020년 1,2회 정보처리기사 필기 기출문제 소프트웨어 개발)

제품 소프트웨어 버전관리

버전 관리

- 제품 소프트웨어 버전 관리는 SW 개발과 관련하여 코드와 라이브러리, 관련 문서 등 시간의 변화에 따른 변경을 관리하는 전체 활동을 의미한다.
- 동일한 소스 코드에 대한 여러 버전을 관리
- 팀 단위로 개발 중인 소스 코드나, 설계도 등의 디지털 문서를 관리
- 문서의 변경 사항들에 숫자나 문자로 이뤄진 "버전"을 부여해서 구분
- 버전을 통해 변경된 시간, 변경된 사항, 변경 작업을 행한자를 파악 할 수 있다.

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

소프트웨어 개발

버전 관리 도구 유형

버전 관리 도구 유형	설 명
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- 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

소프트웨어 개발

현업에서 쓰이는 다양한 버전 관리 도구별 특징

버전 관리 도구	설 명
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.
SVN (Subversion)	<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 버전 컨트롤 툴로서 대규모 프로젝트에서 빠른 속도를 내도록 개발되었다.

소프트웨어 개발

버전 관리 도구	설 명
Git	<ul style="list-style-type: none"> - 기존 리눅스 커널의 버전 컨트롤을 하는 Bitkeeper를 대체 하기 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 효과적 이고 실제로 유용하다. - Git는 SVN과 다르게 Commit은 로컬 저장소에서 이루어 지고 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. <p><주요 특징></p> <ol style="list-style-type: none"> 1) Git의 작업 폴더는 모두 전체 기록과 각 기록을 추적할 수 있는 정보를 포함하는 완전한 형태의 저장소 2) 네트워크에 접근하거나 중앙 서버에 의존하지 않음.
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.

소프트웨어 개발

버전 관리 도구를 활용한 통합 버전 현황 관리

관리 작업	활용사례
버전 관리, 백업 및 복구	<ul style="list-style-type: none">- 이전 버전으로의 복구 필요시- 이전 버전의 기능을 재활용 시
동일 버전 공동 작업	<ul style="list-style-type: none">- 동일 버전의 소스 코드를 두 명 이상의 개발자가 수정할 경우- 분산환경에서 두 명 이상의 개발자가 동일한 버전의 소스를 수정할 경우
여러 버전 솔루션 작업	<ul style="list-style-type: none">- 하나의 솔루션을 기반으로 여러 버전의 제품을 개발 할 경우

소프트웨어 개발

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- 백업결과는 버전 관리자 주기적으로 점검하며, 그 결과를 버전 관리 정기 보고서 함께 보고

버전 관리 도구를 활용한 소스 및 자료 백업

- 디렉토리를 신규로 만들며 물리적 복사: Repository Backup
- 버전 관리 도구의 백업 기능 활용: Git bundle

문제풀이

- 소프트웨어 형상 관리와 가장 거리가 먼 것은?

- ① 제품 측정
- ② 감사 추적 기록
- ③ 변경 관리
- ④ 버전 관리

(2015_7급_국가직_소프트웨어공학)

소프트웨어 개발

빌드 자동화 도구

- 빌드란 소프트웨어를 생성, 테스트, 검사하여 배포하기 위한 모든 것
- 소스코드 파일들을 실행할 수 있는 소프트웨어로 변환하는 과정에 필요한 도구

Ant	<ul style="list-style-type: none">• 아파치 소프트웨어 재단에서 개발한 소프트웨어• XML 기반의 빌드 스크립트를 사용하며 자유도와 유연성이 높다.
Maven	<ul style="list-style-type: none">• Ant의 대안으로 개발됨.• 자동으로 필요한 라이브러리 파일을 불러온다.
Gradle(그레들)	<ul style="list-style-type: none">• 기존의 Ant와 Maven을 보완하여 개발된 빌드 도구• Task 는 작업의 최소단위이며 태스크 단위로 실행한다.• Groovy를 기반으로 하며 안드로이드 스튜디오의 공식 빌드 시스템.
Jenkins(젠킨스)	<ul style="list-style-type: none">• 자바기반의 오픈소스, 가장 많이 활용되는 빌드 자동화 도구임.• 소프트웨어 개발시 지속적 통합(continuous integration) 서비스를 제공하는 툴이다.

문제풀이

- 빌드 자동화 도구에 대한 설명으로 틀린 것은?
 - ① Gradle은 실행할 처리 명령들을 모아 태스크로 만든 후 태스크 단위로 실행한다.
 - ② 빌드 자동화 도구는 지속적인 통합개발환경에서 유용하게 활용된다.
 - ③ 빌드 자동화 도구에는 Ant, Gradle, Jenkins등이 있다.
 - ④ Jenkins는 Groovy 기반으로 한 오픈소스로 안드로이드 앱 개발 환경에서 사용된다.

(2020년 4회 정보처리기사 필기 기출문제 소프트웨어 개발)