

제품 소프트웨어 패키징

1. 제품 소프트웨어 패키징

(1) 제품 소프트웨어 패키징

▼ 개념

- 개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 포장하는 과정

▼ 특성

- 전체 내용 포함하는 매뉴얼 작성
- 버전 관리 / 릴리즈 노트 - 지속적 관리
- 고객 중심 - 고객 편의성 중심
- 모듈화 - 신규 및 변경 개발 소스를 식별, 이를 모듈화하여 사용 제품으로 패키징

▼ 모듈 빌드

- 제품 소프트웨어 패키징은 모듈들을 통해 수행되므로, 최초 소스 개발 시 해당 모듈 단위 및 모듈화를 통해 제품 소프트웨어 패키징을 수행

▼ 모듈화

- 모듈(기능 단위로 분해, 추상화되어 재사용 및 공유 가능한 수준으로 만들어진 단위)을 이용하여 소프트웨어의 성능을 향상, 시스템의 디버깅, 시험, 통합 및 수정을 용이하도록 하는 소프트웨어 설계 기법
- 결합도 최소화, 응집도 최대화
- 개발 편의성 증대(오류 파급효과 최소화), 복잡성 감소(기능 분리, 인터페이스 단순)

▼ 제품 소프트웨어 패키징

- 모듈의 개념을 정확히 파악 후 기능 단위로 패키징하는 것이 필요
- 배포 전 시험 및 수정 등의 작업 진행에서도 모듈 단위로 분류하여 작업

▼ 사용자 중심의 패키징 작업

- 고객 편의성 및 사용자 실행 환경 우선 고려
- 시스템 환경, 진관적 UI, 관리 서비스, 안정적 배포

▼ 과정

- 기능 식별
- 모듈화
- 빌드 진행
- 사용자 환경 분석
- 패키징 적용 시험
- 패키징 변경 개선

(2) 릴리즈 노트

- 고객과 잘 정리된 릴리즈 정보를 공유하는 문서
- 상세 서비스 포함, 회사가 제공하는 제품을 만들어 수정, 변경 또는 개선하는 일련의 작업들, 릴리즈 정보들이 이러한 문서를 통해 제공됨

▼ 중요성

- 정보 제공
- 관리의 용이성 - 전체적인 버전 관리 및 릴리즈 정보를 체계적으로 관리

▼ 고려사항

- 개발 팀에서 직접 작성
- 완전한 정보
- 신규 소스, 빌드 등 이력을 정확히 관리
- 최초 및 변경, 개선 항목까지 연결되어 다음 항목에 대한 정보들이 릴리즈 노트를 통해 작성

▼ 항목

- 소프트웨어 릴리즈의 최근 변경 사항, 개선 사항 및 버그 수정을 간결히 요약

- 헤더, 개요, 목적, 이슈 요약, 재현 항목, 수정/개선 내용, 사용자 영향도, 소프트웨어 지원 영향도, 노트, 면책 조항, 연락 정보

▼ 예외 케이스

- 릴리즈 정보의 예외 케이스에 따른 추가 및 개선 항목이 나타날 경우 릴리즈 노트 항목이 추가될 수 있음
- 테스트 단계에서의 베타 버전 출시
- 긴급 버그 수정 시
- 자체 기능 향상을 포함한 모든 추가 기능의 향상
- 사용자 요청에 따른 특이한 케이스 발생

▼ 작성 프로세스

- 모듈 식별
- 릴리즈 정보 확인
- 릴리즈 노트 개요 작성
- 영향도 체크
- 정식 릴리즈 노트 작성
- 추가 개선 항목 식별

(3) 패키징 도구를 활용한 설치, 배포 수행

1) 패키징 도구

- 배포를 위한 패키징 시에 디지털 콘텐츠의 지적 재산을 보호하고 관리하는 기능을 제공, 안전한 유통과 배포를 보장하는 도구이자 솔루션
- 패키징 작업 진행 시 암호화/보안 기능을 고려하여 패키징할 수 있도록 도움
- 기기종 연동 고려
- 사용자 편의성 고려
- 적합한 암호화 알고리즘 적용

(4) 제품 소프트웨어 저작권 보호의 이해와 필요성

▼ 개념

- 독점적 권리로 타인의 침해를 받지않을 고유한 권한

▼ 특성

- 콘텐츠 복제 제한적 허용
- 종량제 - 클리어링 하우스(디지털 저작권 라이선싱을 중개하고 라이선스 발급을 수행하는 정산소)를 통한 이용 시간에 비례한 과금
- 암호화/보안기능

▼ 패키징 도구 활용

▼ 저작권 관리(DRM 기능)

- 중앙의 클리어링하우스에서 콘텐츠 제공자, 분배자, 소비자 간의 패키징 배포 및 키 관리, 라이선스 발급 관리를 수행

▼ 암호화/보안 기능

- 패키징 작업 시에 도구를 이용하여 나오는 결과물에 대해서도 신뢰할 수 있도록 작업
- 암호화, 키 관리, 식별 기술, 저작권 표현, 암호화 파일 생성, 정책 관리, 크랙 방지, 인증

(5) 애플리케이션 배포 도구를 활용한 배포 프로세스

1. 빌드 내용 식별
2. 패키징 도구 식별
3. DRM 흐름 확인하여 패키징 수행
4. 패키징 도구 설치
5. 배포 작업
6. 정상 배포 확인

2. 제품 소프트웨어 매뉴얼 작성

(1) 소프트웨어 매뉴얼

▼ 개념

- 제품 소프트웨어 개발 단계부터 적용한 기준이나 패키징 이후 설치 및 사용자 측면의 주요 내용 등을 기록한 문서
- 사용자 중심의 기능 및 방법을 나타낸 설명서와 안내서
- 설치 매뉴얼, 사용자 매뉴얼

▼ 작성 항목

- 목차 및 개요
- 문서 이력 정보
- 설치 매뉴얼 주석
- 설치 도구 구성

▼ 설치 환경 체크 항목

- 사용자 환경
- 응용 프로그램
- 업그레이드 버전
- 백업 폴더 확인

▼ 구성 요소

설치 방법을 순서대로 상세히 설명, 설치 화면을 캡처하여 사용자 이해 도움

- 제품 소프트웨어 개요
- 설치 관련 파일
- 설치 절차
- 설치 아이콘
- 삭제 방법
- 설치 버전 및 작성자
- 고객 지원 방법 및 FAQ

- 준수 정보&제한 정보

▼ 프로세스

1. 개요 및 기능 식별
2. UI 분류
3. 설치 파일/백업 파일 확인
4. 삭제 절차 확인
5. 이상 유형 확인
6. 최종 매뉴얼 적용

(2) 사용자 매뉴얼

▼ 개념

- 개발이 완료된 제품 소프트웨어를 고객에게 전달하기 위한 형태로 패키징하고, 설치와 사용에 필요한 제반 절차 및 환경 등 전체 내용을 포함하는 매뉴얼
- 패키지 기능, 패키지 인터페이스, 포함하고 있는 메서드나 오퍼레이션과 메서드의 파라미터 등의 설명 포함

▼ 작성 항목

- 목차 및 개요
- 문서 이력 정보
- 사용자 매뉴얼 주석
- 기록 항목
- 기본 사항
- 고객 지원 방법 및 FAQ
- 준수 정보 & 제한 보증

▼ 프로세스

1. 작성 지침 정의

2. 사용자 매뉴얼 구성요소 정의
3. 구성요소 별 내용 작성
4. 사용자 매뉴얼 검토

(3) 제품 소프트웨어 배포본

▼ 개념

- 배포 정보를 포함한 개발된 컴포넌트 또는 패키지가 제품화된 문서
- 버전, 시스템이 설치 및 운영을 위한 요구사항, 설치 방법, 달라진 기능, 알려진 버그 및 대처 방법 등을 포함하여 배포

▼ 고려사항

- 안정성
- 저작권 및 보안
- 고유 시리얼 넘버

(4) 제품 소프트웨어 매뉴얼의 배포용 미디어 제작

▼ 포함 항목

- 버전 정보, 요구 사양, 설치 방법, 새로운 기능, 알려진 오류/대처 방법, 제약 사항

▼ 제작 방법

- 오프라인 미디어

1. 오프라인 미디어 준비
2. 시리얼 넘버 확인
3. 매뉴얼 확인
4. Install 파일 확인
5. 제작, 정상 동작 확인

- 온라인 미디어

실행 파일로 통합하여 제작

기능 업그레이드, 제품 패치, 기능적 오류 수정, OS 등의 업그레이드에 따른 호환성 제공

▼ 과정

1. 배포용 미디어 선정
2. 관리 체계 확인
3. 설치 파일/매뉴얼 확인
4. 최종 배포본 검증
5. 배포용 미디어 정보 확인
6. 최종 미디어 인증확인 및 배포

3. 제품 소프트웨어 버전 등록

(1) 제품 소프트웨어 패키징 형상 관리

▼ 개념

- 소프트웨어 변경 사항을 체계적으로 추적하고 통제하는 관리 기법
- 변경 통제 시점을 정하고, 변경을 관리 통제, 이에 따라 전체적인 버전 체계가 관리

▼ 중요성

- 지속적으로 변경되는데 이에 대한 개발 통제가 중요
- 배포판의 버그 및 수정에 대한 추적의 결여 및 무절제한 변경이 난무할 수 있음
- 제품 소프트웨어의 가시성 결핍 발생

▼ 역할

- 관리 유용
- 동시 개발
- 빠른 복구
- SW 적시 공급

(2) 버전 관리

- SW 개발과 관련하여 코드와 라이브러리, 관련 문서 등 시간의 변화에 따른 변경을 관리하는 전체 활동
- 최종 인증을 패키징 버전에 한하여 컴포넌트 저장소에 등록

▼ 항목

- Import, Add, Check-out(저장소 파일 받기), Check-in(새로운 버전 갱신), Update, Commit, Repository, Diff

(3) 버전 관리 도구

▼ 개념

- 형상 관리 지침을 활용하여 제품 소프트웨어의 신규 개발, 변경, 개선과 관련된 수정 사항을 관리하는 도구

▼ 유형

- 공유 폴더 방식

매일 개발이 완료된 파일은 약속된 위치의 공유 폴더에 복사하는 방식

RCS, SCCS

- 클라이언트/서버 방식

중앙에 버전 관리 시스템을 항상 동작

현재 작업 내용과 이전 작업내용 추적에 용이

서로 다른 개발자가 같은 파일을 작업했을 때 경고 메시지 출력

CVS, SVN

- 분산 저장소 방식

로컬 저장소와 원격 저장소로 분리된 구조

개발 완료한 파일을 수정한 다음에 로컬 저장소에 우선적으로 커밋한 이후 다시 원격 저장소에 반영하는 방식

▼ 유의사항

- 버전에 대한 쉬운 정보 접근성
- 불필요한 사용자에게 대한 접근 제어
- 동일 프로젝트에 대한 동시 사용성

동일 프로젝트 여러 개발자가 동시 개발

여러 개발자가 동일한 파일, 폴더에 접근 시 동시에 파일 수정이 일어나더라도 개발자의 수정 내역이 통합될 수 있어야 함

- 빠른 오류 복구

(4) 버전 관리도구 활용

▼ 활용 방향

- 보유 제품 버전과 새로 개발된 버전을 동시에 유지, 관리
- 버전 관리 도구의 상세 기능을 활용하여 새 버전을 유지하면서 기존 버전의 관리도 함께 수행
- 백업 및 장애 복구 등의 추가적인 사항도 Case 별로 고려

▼ 버전 현황 관리 기법

- 기존의 고유한 버전 관리 외에도 버전 관리 도구에서 지원하는 모든 관리 작업 전체를 관리할 수 있어야 함
- 버전별 공동 작업 이외에도 백업 및 복구 작업, 공동 솔루션 기반의 여러 가지 작업 등을 통합, 관리
- 버전 관리 도구를 통한 관리 작업

버전 관리 백업 및 복구

동일 버전 공동 작업

여러 버전 솔루션 작업

▼ 백업

- 문제에 대비해 시스템을 회복 가능한 상태로 만들기 위해 준비
- 손실의 예방 및 최소화 우선