**5과목 정보시스템 구축 관리**

1장 소프트웨어 개발 방법론 활용

166 소프트웨어 개발 방법론

- 소프트웨어 개발, 유지보수 등에 필요한 여러 가지 일들의 수행 방법과 이러한 일들을 효율적으로 수행하려는 과정에서 필요한 각종 기법 및 도구를 체계적으로 정리하여 표준화한 것

구조적 방법론 – 정형화된 분석 절차에 따라 사용자 요구사항을 파악하여 문서화하는 처리 중심의 방법론

절차 : 타당성 검토 단계 -> 계획 단계 -> 요구사항 단계 -> 설계 단계 -> 구현 단계 -> 시험 단계 -> 운용/유지보수 단계

정보공학 방법론 – 정보 시스템의 개발을 위해 계획, 분석, 설계, 구축에 정형화된 기법들을 상호 연관성 있게 통합 및 적용하는 자료 중심의 방법론

절차 : 정보 전략 계획 수립 단계 -> 업무 영역 분석 단계 -> 업무 시스템 설계 단계 -> 업무 시스템 구축 단계

객체지향 방법론 – 객체들을 조립해서 필요한 소프트웨어를 구현하는 방법론

절차 : 요구 분석 단계 -> 설계 단계 -> 구현 단계 -> 테스트 및 검증 단계 -> 인도 단계

컴포넌트 기반 방법론 – 기존의 시스템이나 소프트웨어를 구성하는 컴포넌트를 조합하여 하나의 새로운 애플리케이션을 만드는 방법론

절차 : 개발 준비 단계 -> 분석 단계 -> 설계 단계 -> 구현 단계 -> 테스트 단계 -> 전개 단계 -> 인도 단계

애자일 방법론 – 고객의 요구사항 변화에 유연하게 대응할 수 있도록 일정한 주기를 반복하면서 개발 과정을 진행하는 방법론

절차 : 사용자 스토리 -> 계획 -> 개발 -> 승인 테스트

제품 계열 방법론 – 특정 제품에 적용하고 싶은 공통된 기능을 정의하여 개발하는 방법론

167 S/W 공학의 발전적 추세

소프트웨어 재사용 – 이미 개발되어 인정받은 소프트웨어의 전제 혹은 일부분을 다른 소프트웨어 개발이나 유지에 사용하는 것

합성 중심 – 블록을 만들어서 끼워 맞추어 소프트웨어를 완성시키는 방법 / 블록 구성 방법

생성 중심 – 추상화 형태로 쓰여진 명세를 구체화하여 프로그램을 만드는 방법 / 패턴 구성 방법

소프트웨어 재공학 – 새로운 요구에 맞도록 기존 시스템을 이용하여 보다 나은 시스템을 구축하고 새로운 기능을 추가하여 소프트웨어 성능을 향상시키는 것

분석 / 재구성 / 역공학 / 이식

CASE – 소프트웨어 개발 과정에서 사용되는 요구 분석, 설계, 구현, 검사 및 디버깅 과정 전체 또는 일부를 컴퓨터와 전용 소프트웨어 도구를 사용하여 자동화하는 것

168 비용 산정 기법

- 소프트웨어의 개발 규모를 소요되는 인원, 자원, 기간 등으로 확인하여 실행 가능한 계획을 수립하기 위해 필요한 비용을 산정하는 것

프로젝트 요소 – 제품 복잡도 / 시스템 크기 / 요구되는 신뢰도

자원 요소 – 인적 자원 / 하드웨어 자원 / 소프트웨어 자원

생산성 요소 – 개발자 능력 / 개발 기간

169 비용 산정 기법 – 하향식

- 과거의 유사한 경험을 바탕으로 전문 지식이 많은 개발자들이 참여한 회의를 통해 비용을 산정하는 비과학적인 방법

전문가 감정 기법 – 조직 내에 있는 경험이 많은 두 명 이상의 전문가에게 비용 산정을 의뢰하는 기법

델파이 기법 – 전문가 감정 기법의 주관적인 편견을 보완하기 위해 많은 전문가의 의견을 종합하여 산정하는 기법

170 비용 산정 기법 – 상향식

- 프로젝트의 세부적인 작업 단위별로 비용을 산정한 후 집계하여 전체 비용을 산정하는 방법

LOC 기법 – 소프트웨어 각 기능의 원시 코드 라인 수의 비관치, 낙관치, 기대치를 측정하여 예측치를 구하고 이를 이용하여 비용을 산정하는 기법

예측치 a+4m+b / 6

노력 = 개발 \* 투입 인원 = LOC/1인당 월평균 생산 코드 라인 수

개발 비용 = 노력 \* 단위 비용

개발 기간 = 노력 / 투입 인원

생산성 = LOC / 노력

개발 단계별 인월수 기법 – 각 기능을 구현시키는 데 필요한 노력을 생명 주기의 각 단계별로 산정

171 수학적 산정 기법

- 상향식 비용 산정 기법으로 경험적 추정 모형, 실험적 추정 모형, 개발 비용 산정의 자동화를 목표로 함

COCOMO 모형 / Putnam 모형 / 기능 점수(FP) 모형

COCOMO – 원시 프로그래므이 규모인 LOC에 의한 비용 산정 기법임

-> 조직형 : 기관 내부에서 개발된 중,소 규모의 소프트웨어로 일괄 자료 처리나 과학 기술 계산용, 비즈니스 자료 처리용으로 5만 라인 이하의 소프트웨어를 개발하는 유형

노력 = 2.4 \*

개발 기간 = 2.5 \*

-> 반분리형 : 트랜잭션 처리 시스템이나 운영체제, 데이터베이스 관리 시스템 등의 30만 라인 이하의 소프트웨어를 개발하는 유형

노력 = 3.1 \*

개발 기간 = 2.5 \*

-> 내장형 : 초대형 규모의 트랜잭션 처리 시스템이나 운영체제 등의 30만 라인 이상의 소프트웨어를 개발하는 유형

노력 = 3.6 \*

개발 기간 = 2.5 \*

Putnam 모형 – 소프트웨어 생명 주기의 전 과정 동안에 사용될 노력의 분포를 가정해주는 모형

개발 노력 =

기능 점수 모형 – 소프트웨어의 기능을 중대시키는 요인별로 가중치를 부여하고 요인별 가중치를 합산하여 총 기능 점수를 산출하며 총 기능 점수와 영향도를 이용하여 기능 점수를 구한 후 비용을 산정하는 기법

기능 점수 = 총 기능 점수 \* [0.65 + (0.1 \* 총 영향도)]

172 프로젝트 일정 계획

- 프로젝트의 프로세스를 이루는 소작업을 파악하고 예측된 노력을 각 소작업에 분배하며 소작업의 순서와 일정을 정하는 것

WBS / PERT/CPM / 간트 차트

PERT – 프로젝트에 필요한 전체 작업의 상호 관계를 표시하는 네트워크로 각 작업별로 낙관적인 경우, 가능성이 있는 경우, 비관적인 경우로 나누어 각 단계별 종료 시기를 결정하는 방법

작업 예측치 = 평방 편차 =[

CPM – 프로젝트 완성에 필요한 작업을 나열하고 작업에 필요한 소요 기간을 예측하는데 사용하는 기법

간트 차트 – 프로젝트의 각 작업들이 언제 시작하고 언제 종료되는지에 대한 작업 일정을 막대 도표를 이용하여 표시하는 프로젝트 일정표로 시간선 차트라고도 함

173 소프트웨어 개발 방법론 결정

- 프로젝트 관리와 재사용 현황을 소프트웨어 개발 방법론에 반영하고 확정된 소프트웨어 생명 주기와 개발 방법론에 맞춰 소프트웨어 개발 단계, 활동, 작업, 절차 등을 정의하는 것

소프트웨어 개발 방법론 결정 절차

-> 프로젝트 관리와 재사용 현황을 소프트웨어 개발 방법론에 반영

-> 개발 단계별 작업 및 절차를 소프트웨어 생명 주기에 맞춰 수립

-> 결정된 소프트웨어 개발 방법론의 개발 단계별 활동 목적, 작업 내용, 산출물에 대한 매뉴얼 작성

174 소프트웨어 개발 표준

- 소프트웨어 개발 단계에서 수행하는 품질 관리에 사용되는 국제 표준을 의미

ISO/IEC 12207 : 국제표준에서 만든 표준 소프트웨어 생명 주기 프로세스

기본 생명 주기 프로세스 / 지원 생명 주기 프로세스 / 조직 생명 주기 프로세스

CMMI – 소프트웨어 개발 조직의 업무 능력 및 조직의 성숙도를 평가하는 모델

성숙도 – 초기 / 관리 / 정의 / 정량적 관리 / 최적화

SPICE – 정보 시스템 분야에서 소프트웨어의 품질 및 생산성 향상을 위해 소프트웨어 프로세스를 평가 및 개선하는 국제 표준 공식 명칙 ISO/IEC 15504

단계 – 불완전 / 수행 / 관리 / 확립 / 예측 / 최적화

범주 – 고객-공급자 / 공학 프로세스 / 지원 프로세스 / 관리 프로세스 / 조직 프로세스

175 소프트웨어 개발 방법론 테일러링

- 프로젝트 상황 및 특성에 맞도록 정의된 소프트웨어 개발 방법론의 절차, 사용기법 등을 수정 및 보완하는 작업

절차 : 프로젝트 특징 정의 -> 표준 프로세스 선정 및 검증 -> 상위 수준의 커스터마이징 -> 세부 커스터마이징 -> 테일러링 문서화

고려사항 : 내부적 기준 – 목표 환경 / 요구사항 / 프로젝트 규모 / 보유 기술

외부적 기준 – 법적 제약사항 / 표준 품질 기준

소프트웨어 개발 방법론 테일러링 기법

-> 프로젝트 규모와 복잡도에 따른 테일러링 기법

-> 프로젝트 구성원에 따른 테일러링 기법

-> 팀내 방법론 지원에 따른 테일러링 기법

-> 자동화에 따른 테일러링 기법

176 소프트웨어 개발 프레임워크

- 소프트웨어 개발에 공통적으로 사용되는 구성 요소와 아키텍처를 일반화하여 손쉽게 구현할 수 있도록 여러 가지 기능들을 제공해주는 반제품 형태의 소프트웨어 시스템

스프링 프레임워크 – 자바 플랫품을 위한 오픈 소스 경량형 애플리케이션 프레임워크

전자정부 프레임워크 – 우리나라의 공공부문 정보화 사업 시 효율적인 정보 시스템의 구축을 지원하기 위해 필요한 기능 및 아키텍처를 제공하는 프레임워크

닷넷 프레임워크 – Windows 프로그램의 개발 및 실행 환경을 제공하는 프레임워크, 통합 인터넷 전략을 위해 개발됨

2장 IT 프로젝트 정보시스템 구축 관리

177 네트워크 관련 신기술

지능형 초연결망 – 스마트 시티, 스마트 스테이션 등 4차 산업혁명 시대를 맞아 새로운 변화에 따라 급격하게 증가하는 데이터 트래픽을 효과적으로 수용하기 위해 시행되는 정부 주관 사업

소프트웨어 정의 기술 – 네트워크, 데이터 센터 등에서 소유한 자원을 가상화하여 개별 사용자에게 제공하고 중앙에서는 통합적으로 제어가 가능한 기술

소프트웨어 정의 네트워킹 / 소프트웨어 정의 데이터센터 / 소프트웨어 정의 스토리지

IoT – 정보 통신 기술을 기반으로 실세계와 가상 세계의 다양한 사물들을 인터넷으로 서로 연결하여 진보된 서비스를 제공하기 위한 서비스 기반 기술

IoT 관련 용어

M2M / 메시 네트워크 / 와이선 / UWB / 피코넷 / USN / SON / 저전력 블루투스 기술 / NFC

클라우드 컴퓨팅 – 각종 컴퓨팅 자원을 중앙 컴퓨터에 두고 인터넷 기능을 갖는 단말기로 언제 어디서나 인터넷을 통해 컴퓨터 작업을 수행할 수 있는 환경

모바일 클라우드 컴퓨팅 / 인터클라우드 컴퓨팅 / 클라우드 기반 HSM / 파스-타

기타 용어

Zing / NDN / NGN / 올-IP / WBAN / GIS / 애드 혹 네트워크 / 네트워크 슬라이싱 / 파장 분할 다중화 / 개방형 링크드 데이터 / SSO / 스마트 그리드

178 네트워크 구축

통신망 – 정보를 전달하기 위해서 통신 규약에 의해 연결한 통신 설비의 집합

성형 – 중앙에 중앙 컴퓨터가 있고 이를 중심으로 단말장치들이 연결되는 중앙 집중식의 네트워크 구성 형태

링형 – 컴퓨터와 단말장치들을 서로 이웃하는 것끼리 포인트 투 포인트 방식으로 연결시킨 형태

버스형 – 한 개의 통신 회선에 여러 대의 단말 장치가 연결되어 있는 형태

계층형 – 중앙 컴퓨터와 일정 지역의 단말장치까지는 하나의 통신 회선으로 연결시키고 이웃하는 단말장치는 일정 지역 내에 설치된 중간 단말장치로부터 다시 연결시키는 형태

망형 – 모든 지점의 컴퓨터와 단말장치를 서로 연결한 형태

네트워크 – LAN / WAN

179 스위치

- 브리지와 같이 LAN과 LAN을 연결하여 훨씬 더 큰 LAN을 만드는 장치로 ISI 7계층의 레이어를 따라 L2, L3, L4, L7로 분류

L2 스위치 – OSI 2계층에 속하는 장비 / MAC 주소를 기반으로 프레임을 전송

L3 스위치 – OSI 3계층에 속하는 장비 / IP 주소를 기반으로 패킷을 전송

L4 스위치 – OSI 4계층에 속하는 장비 / IP 주소 및 TCP/UDP를 기반으로 사용자들의 요구를 서버의 부하가 적은 곳에 배분하는 로드밸런싱 기능을 제공

L7 스위치 – OSI 7계층에 속하는 장비 / IP주소, TCP/UDP 포트 정보에 패킷 내용까지 참조하여 세밀하게 로드밸런싱함

스위칭 – Store and Forwarding / Cut-through / Fragment Free

백본 – 여러 네트워크들을 연결할 때 중추적 역할을 하는 네트워크

백본 스위치 – 백본에서 스위칭 역할을 하는 장비

180 경로 제어 / 트래픽 제어

경로 제어 – 송수신 측 간의 전송 경로 중에서 최적 패킷 교환 경로를 결정하는 기능

경로 제어 요소 : 성능 기준, 경로의 결정 시간과 장소, 정보 발생지, 경로 정보의 갱신 시간

경로 제어 프로토콜 – 효율적인 경로 제어를 위해 네트워크 정보를 생성, 교환, 제어하는 프로토콜을 총칭

IGP / EGP / BGP

흐름 제어 – 네트워크 내의 원활한 흐름을 위해 송수신 측 사이에 전송되는 패킷의 양이나 속도를 규제하는 기능

정지-대기 / 슬라이딩 윈도우

폭주(혼잡)제어 – 흐름제어가 송수신 측 사이의 패킷 수를 제어하는 기능이라면 폭주 제어는 네트워크 내의 패킷 수를 조절하여 네트워크를 오버플를 방지하는 기능

느린 시작 / 혼잡 회피

교착 상태 방지 – 교착상태란 교환기 내에 패킷들을 축적하는 기억 공간이 꽉 차 있을 때 다음 패킷들이 기억 공간에 들어가기 위해 무한정 기다리는 현상을 말함

181 SW 관련 신기술

인공지능 – 인간의 두뇌와 같이 컴퓨터 스스로 추론, 학습, 판단 등 인간지능적인 작업을 수행하는 시스템

뉴럴링크 – 작은 전극을 뇌에 이식함으로써 생각을 업로드하고 다운로드하는 것을 목표로 삼음

딥 러닝 – 인간의 두뇌를 모델로 만들어진 인공 신경망을 기반으로 하는 기계 학습 기술

전문가 시스템 – 의료 진단 등과 같은 특정 분야의 전문가가 수행하는 고도의 업무를 지원하기 위한 컴퓨터 응용 프로그램

가상현실 – 컴퓨터 등을 사용하여 실제와 유사하지만 실제가 아닌 환경이나 상황을 구현하는 기술

증강현실 – 실제 촬영한 화면에 가상의 정보를 부가하여 보여주는 기술

혼합현실 – 가상현실과 현실세계를 합쳐, 현실의 물리적인 객체와 가상의 객체가 상호작용할 수 있는 환경을 구현하는 기술

그레이웨어 – 악의적이지 않은 유용한 소프트웨어라고 할 수도 있고 악의적일 수도 있다

매시업 – 웹에서 제공하는 정보 및 서비스를 이용하여 새로운 소프트웨어나 서비스, 데이터베이스 등을 만드는 기술

시멘틱 웹 – 컴퓨터가 사람을 대신하여 정보를 읽고 이해하고 가공하여 새로운 정보를 만들어 낼 수 있도록 이해하기 쉬운 의미를 가진 차세대 지능형 웹이다

증발품 – 판매 계획 또는 배포 계획은 발표되었으나 실제로 고객에게 판매되거나 배포되지 않고 있는 소프트웨어

오픈 그리드 서비스 아키텍처 – 애플리케이션 공유를 위한 웹 서비스를 그리드 상에서 제공하기 위해 만든 개방형 표준

서비스형 소프트웨어 – 사용자가 필요로 하는 서비스만 이용할 수 있도록 한 소프트웨어

복잡 이벤트 처리 – 실시간으로 발생하는 많은 사건들 중 의미가 있는 것만을 추출할 수 있도록 사건 발생 조건을 정의하는 데이터 처리 방법

텐서플로 – 다양한 작업에 대해 데이터 흐름 프로그래밍을 위한 오픈소스 소프트웨어 라이브러리

도커 – 컨테이너 기술을 자동화하여 쉽게 사용할 수 있게 하는 오픈소스 프로젝트

스크래피 – 코드 재사용성을 높이는 데 도움이 됨

182 보안 관련 신기술

블록체인 / 분산 원장 기술 / 양자 암호키 분배 / 프라이버시 강화 기술 / 공통 평가 기준 / 개인정보 영향평가 제도 / 소프트웨어 에스크로 / 서비스형 블록체인 / OWASP / TCP 래퍼 / 허니팟 / DPI

183 소프트웨어 개발 보안 관련 기관과 직무별 보안 활동

소프트웨어 개발 보안 – 보안 취약점을 최소화하여 보안 위협으로부터 안전한 소프트웨어를 개발하기 위한 일련의 보안 활동

소프트웨어 개발 보안과 관련 기관 – 행정안전부, 행정기관, 한국인터넷진흥원, 사업자, 감리법인

소프트웨어 개발 직무별 보안활동 - 프로젝트 관리자, 요구사항 분석가, 아키텍트, 설계자, 구현 개발자, 테스트 분석가, 보안 감사자

184 소프트웨어 개발 보안 활동 관련 법령 및 규정

개인정보 보호 관련 법령

개인정보 보호법 / 정보통신망 이용촉진 및 정보보호 등에 관한 법률 / 신용정보의 이용 및 보호에 관한 법률 / 위치정보의 보호 및 이용 등에 관한 법률 / 표준 개인정보 보호 지침 / 개인정보의 안전성 확보 조치 기준 / 개인정보 영향평가에 관한 고시

IT 기술 관련 규정

RFID 프라이버시 보호 가이드라인 / 위치정보의 보호 및 이용 등에 관한 법률 / 위치정보의 관리적, 기술적 보호조치 권고 해설서 / 바이오정보 보호 가이드라인 / 뉴미디어 서비스 개인정보 보호 가이드라인

185 HW 관련 신기술

고가용성 솔루션 – 긴 시간동안 인정적인 서비스 운영을 위해 장애 발생 시 즉시 다른 시스템으로 대체 가능한 환경을 구축하는 메커니즘

3D Printing – 대상을 평면에 출력하는 것이 아니라 손으로 만질 수 있는 실제 물체로 만들어내는 것

4D Printing – 특정 시간이나 환경 조건이 갖추어지면 스스로 형태를 변화시키거나 제조되는 자가 조립 기술이 적용된 제품을 3D Printing하는 기술을 의미

4K 해상도 – 차세대 고화질 모니터의 해상도를 지칭하는 용어

앤 스크린 - N개의 서로 다른 단말기에서 동일한 콘텐츠를 자유롭게 이용할 수 있는 서비스

컴패니언 스크린 – TV 방송 시청 시 방송 내용을 공유하여 추가적인 기능을 수행할 수 있는 스마트폰, 태블릿 PC 등을 의미

신 클라이언트 PC – 하드디스크나 주변 장치 없이 기본적인 메모리만 갖추고 서버와 네트워크로 운용되는 개인용 컴퓨터를 말하는 것

패블릿 – 폰과 태블릿의 합성어 태블릿 기능을 포함한 5인치 이상의 대화면 스마트폰

멤스 – 초정밀 반도체 제조 기술을 바탕으로 센서, 액추에이터 등 기계 구조를 다양한 기술로 미세 가공하여 전기기계적 동작을 할 수 있도록 한 초미세 장치

엠디스크 – 한 번의 기록만으로 자료를 영구 보관할 수 있는 광 저장 장치

멤리스터 – 메모리와 레지스터의 합성어로 전류의 방향과 양등 기존의 경험을 모두 기억하는 특별한 소자

186 Secure OS

- 기존의 운영체제에 내재된 보안 취약점을 해소하기 위해 보안 기능을 갖춘 커널을 이식하여 외부의 침입으로부터 시스템 자원을 보호하는 운영체제를 의미

암호적 분리 / 논리적 분리 / 시간적 분리 / 물리적 분리

Secure OS 보안 기능

식별 및 인증 / 임의적 접근통제 / 강제적 접근통제 / 객체 재사용 보호 / 완전한 조정 / 신뢰 경로 / 감사 및 감사기록 축소

187 DB 관련 신기술

빅데이터 – 기존의 관리 방법이나 분석 체계로는 처리하기 어려운 막대한 양의 정형 또는 비정형 데이터 집합

브로드 데이터 – 다양한 채널에서 소비자와 상호 작용을 통해 생성된 기업 마케팅에 있어 효율적이고 다양한 데이터이며 기존데이터에 새로운 가치가 더해진 데이터를 말함

메타 데이터 – 일련의 데이터를 정의하고 설명해 주는 데이터

디지털 아카이빙 – 디지털 정보 자원을 장기적으로 보존하기 위한 작업을 말함

하둡 – 오픈 소스를 기반으로 한 분산 컴퓨팅 플랫폼

맵리듀스 – 대용량 데이터를 분산 처리하기 위한 목적으로 개발된 프로그래밍 모델

타조 – 오픈 소스 기반 분산 컴퓨팅 플랫폼인 아파치 하둡 기반의 분산 데이터 웨어하우서 프로젝트

데이터 다이어트 – 데이터를 삭제하는 것이 아닌 압축하고 중복된 정보는 중복을 배제하고 새로운 기존에 따라 나누어 저장하는 작업

데이터 마이닝 – 데이터 웨어하우스에 저장된 데이터 집합에서 사용자 요구에 따라 유용하고 가능성 있는 정보를 발견하기 위한 기법

OLAP – 다차원으로 이루어진 데이터로부터 통계적인 요약 정보를 분석하여 의사결정에 활용하는 방식을 말함

188 회복 / 병행제어

회복 – 트랜잭셕들을 수행하는 도중 장애가 발생하여 데이터베이스가 손상되었을 때 손상되기 이전의 정상 상태로 복구하는 작업

장애의 유형 - 트랜잭션 장애 / 시스템 장애 / 미디어 장애

회복기법 – 연기 갱신 기법 / 즉각 갱신 기법 / 그림자 페이지 대체 기법 / 검사점 기법

병행제어 – 다중 프로그램의 이점에 활용하여 동시에 여러 개의 트랜잭션을 병행수행할 때 동시에 실행되는 트랜잭션들이 데이터베이스의 일관성을 파괴하지 않도록 트랜잭션 간의 상호 작용을 제어하는 것

병행제어 기법의 종류 – 로킹 / 타임 스탬프 순서 / 최적 병행수행 / 다중 버전 기법

병행수행의 문제점 – 갱신 분실 / 비완료 의존성 / 모순성 / 연쇄 복귀

189 교착상태

- 상호 배제에 의해 나타나는 문제점으로 둘 이상의 프로세스들이 자원을 점유한 상태에서 서로 다른 프로세스가 점유하고 있는 자원을 요구하며 무한정 기다리는 현상을 의미

교착 상태의 발생 필요 충분 조건 – 상호 배제 / 점유와 대기 / 비선점 / 환형 대기

교착 상태의 해결 방법 – 해방 기법 / 회피 기법 / 발견 기법 / 회복 기법

190 데이터 표준화

- 시스템을 구성하는 데이터 요소의 명칭, 정의, 형식, 규칙에 대한 원칙을 수립하고 적용하는 것을 의미

데이터 표준 – 데이터 모델이나 DB에서 정의할 수 있는 모든 오브젝트를 대상으로 데이터 표준화를 수행해야 함

데이터 표준의 종류

표준 단어 : 업무에서 사용하고 일정한 의미를 갖고 있는 최소 단위의 단어를 의미

표준 도메인 : 컬럼을 성질에 따라 그룹핑한 개념

표준 코드 : 선택할 수 있는 값을 정형화하기 위해 기준에 맞게 이미 정의된 코드값

표준 용어 : 단어, 도메인, 코드 표준이 정의되면 이를 바탕으로 표준 용어를 구성함

데이터 관리 조직 – 데이터 표준 원칙이나 데이터 표준의 준수 여부 등을 관리하는 사람들

데이터 표준화 절차

데이터 표준화 요구사항 수집 -> 데이터 표준 정의 -> 데이터 표준 확정 -> 데이터 표준 관리

데이터 표준화 대상 – 데이터 명칭 / 데이터 정의 / 데이터 형식 / 데이터 규칙

3장 소프트웨어 개발 보안 구축

191 Secure SDLC

- 보안상 안전한 소프트웨어를 개발하기 위해 SDLC에 보안 강화를 위한 프로세스를 포함한 것

Secure SDLC의 대표적인 방법론 – CLASP / SDL / Seven Touchpoints

요구사항 분석 단계 – 보안 항목에 해당하는 요구사항을 식별하는 작업을 수행함

설계 단계 – 보안 요구사항들을 소프트웨어 설계서에 반영하고 보안 설계서를 작성

구현 단계 – 표준 코딩 정의서 및 소프트웨어 개발 보안 가이드를 준수하며 설계서에 따라 보안 요구사항들을 구현

테스트 단계 – 설계 단계에서 작성한 보안 설계서를 바탕으로 보안 사항들이 정확히 반영되고 동작되는지 점검

유지보수 단계 – 이전 과정을 모두 수행하였음에도 발생할 수 있는 보안사고들을 식별하고 사고 발생 시 이를 해결하고 보안 패치를 실시

192 세션 통제

세션 – 서버와 클라이언트 연결을 의미

세션 통제 – 세션의 연결과 연결로 인해 발생하는 정보를 관리하는 것

불충분한 세션 관리 – 일정한 규칙이 존재하는 세션 ID 가 발급되거나 타임아웃이 너무 길게 설정되어 있는 경우 발생할 수 있는 보안 약점

193 입력 데이터 검증 및 표현

- 입력 데이터로 인해 발생하는 문제들을 예방히기 위해 구현 단계에서 검증해야 하는 보안 점검 항목들

SQL 삽입 / 경로 조작 및 자원 삽입 / 크로스사이트 스크립팅 / 운영체제 명령어 삽입 / 위험한 형식 파일 업로드 / 신뢰되지 않는 URL 주소로 자동접속 연결 / 메모리 버퍼 오버플로

194 보안기능

- 소프트웨어 개발의 구현 단계에서 코딩하는 기능인 인증, 접근제어, 기밀성, 암호화 등을 올바르게 구현하기 위한 보안 점검 항목들

보안 약점

적절한 인증 없이 중요 기능 허용 / 부적절한 인가 / 중요한 자원에 대한 잘못된 권한 설정 / 취약한 암호화 알고리즘 사용 / 중요정보 평문 저장 및 전송 / 하드코드 된 비밀번호

195 시간 및 상태

- 동시 수행을 지원하는 병렬 처리 시스템이나 다수의 프로세스가 동작하는 환경에서 시간과 실행 상태를 관리하여 시스템이 원활하게 동작되도록 하기 위한 보안 검증 항목들

TOCTOU 경쟁 조건 – 검사 시점과 사용 시점을 고려하지 않고 코딩하는 경우 발생하는 보안 약점

반복문이나 재귀함수에서 종료 조건을 정의하지 않았거나 논리 구조상 종료될 수 없는 경우 발생하는 보안 약점

196 에러처리

에러처리 – 소프트웨어 실행 중 발생할 수 있는 오류들을 사전에 정의하여 오류로 인해 발생할 수 있는 문제들을 예방하기 위한 보안 점검 항목

오류 메시지를 통한 정보노출 – 오류 발생으로 실행 환경, 사용자 정보, 디버깅 정보 등의 중요 정보를 소프트웨어가 메시지로 외부에 노출하는 보안 약점

오류 상황 대응 부재 – 소프트웨어 개발 중 예외처리를 하지 않았거나 미비로 인해 발생하는 보안 약점

197 코드 오류

코드 오류 – 소프트웨어 구현 단계에서 개발자들이 코딩 중 실수하기 쉬운 형 변환, 자원 변화 등의 오류를 예방하기 위한 보안 점검 항목들

널 포인터 역참조 – 널 포인터가 가리키는 메모리에 어떠한 값을 저장할 때 발생하는 보안 약점

부적절한 자원 해제 – 자원을 반환하는 코드를 누락하거나 프로그램 오류로 할당된 자원을 반환하지 못했을 때 발생하는 보안 약점

해제된 자원 사용 – 이미 사용이 종료되어 반환된 메모리를 참조하는 경우 발생하는 보안 약점

초기화되지 않은 변수 사용 – 변수 선언 후 값이 부여되지 않은 변수를 사용할 때 발생하는 보안 약점

198 캡슐화

- 정보 은닉이 필요한 중요한 데이터와 기능을 불충분하게 캡슐화하거나 잘못 사용함으로써 발생할 수 있는 문제를 예방하기 위한 보안 점검 항목들

잘못된 세션에 의한 정보 노출 – 다중 스레드 환경에서 멤버 변수에 정보를 발생하는 보안 약점

제거되지 않고 남은 디버그 코드 – 개발 중에 버그 수정이나 결과값 확인을 위해 남겨둔 코드들로 인해 발생하는 보안 약점

시스템 데이터 정보 노출 – 시스템의 내부 정보를 시스템 메시지 등을 통해 외부로 출력하도록 코딩했을 때 발생하는 보안 약점

public 메소드로부터 반환된 private 배열 - 선언된 클래스 내에서만 접근이 가능한 private 배열을 모든 클래스에서 접근이 가능한 public 메소드에서 반환할 때 발생하는 보안 약점

199 API 오용

- 소프트웨어 구현 단계에서 API를 잘못 사용하거나 보안에 취약한 API를 사용하지 않도록 하기 위한 보안 검증 항목들

DNS Lookup에 의존한 보안 결정 – 도메인명에 의존하여 인증이나 접근 통제 등의 보안 결정을 내리는 경우 발생하는 보안 약점

취약한 API 사용 – 보안 문제로 사용이 금지된 API를 사용하거나 잘못된 방식으로 API를 사용했을 때 발생하는 보안 약점

200 암호 알고리즘

암호 알고리즘 – 패스워드, 주민번호, 은행계좌와 같은 중요정보를 보호하기 위해 평문을 암호화된 문장으로 만드는 절차 또는 방법을 의미

개인키 암호화 기법 – 동일한 키로 데이터를 암호화하고 복호화함

블록 암호화 방식 : DES / SEED / AES / ARIA / IDEA

스트림 암호화 방식 : LFSR / RC4

공개키 암호화 기법 – 데이터를 암호화할 때 사용하는 공개키는 데이터 베이스 사용자에게 공개하고 복호화할 때의 비밀키는 관리자가 비밀리에 관리

RSA

해시 – 임의의 길이의 입력 데이터나 메시지를 고정된 길이의 값이나 키로 변환하는 것을 의미

SHA / MD5 / N-NASH / SNEFRU