**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