<Chapter3. 프로젝트 관리와 계획>

프로젝트 관리

-목적 : 작업수행에 필요한 자원,인력등을 가장 효과적으로 사용하여 프로젝트 목표 달성.

-어려움 : 개발 대상이 눈에 안 보임, 기술 발전 맹 빠름, 조직마다 프로세스 다름.

-프로젝트 관리 활동의 요소 : 계획, 조직, 모니터링, 조정(제어)

프로젝트시작->계획->실행,모니터링(피드백발생)->종료(보고서나옴)

프로젝트 시작

-목표세우고 가치가 리스크를 이해

-결정요인 : 프로젝트가 제공할 가치, 관련 리스크

-가치평가법 : 금액적측면(투자 회수시간, ROI, 순수현재가치), 비금액적측면(평가표, SWOT)

리스크와 타당성

-위험요소 : 시간, 기술적 어려움, 자원+현재기용성+예상 사용량과 가용성+프로젝트의 우선순위 및 중요도

-타당성분석

1. SOW : 프로젝트가 성취해야할일

2. 비즈니스 목표 : 프로젝트 결과물

3. 예산 : 비용과 수익의 요약

4. 프로젝트 일정 : 대략적 일정

5. 프로젝트 리스크 : 위험요소

6. 대안 : 구축, 구매들의 방법

7. 평가 : 프로젝트 가치에 대한 평가 결과

프로젝트 계획과 스케줄링

-초기계획 : 목표설정, 일정 정의, 비용 추정, p.93그림 참고

-프로젝트 범위 정하기 : 1. 넓은 범위(시간,돈 많이들어감) 2. 작은 범위(별도의 시스템으로 개발)

-WBS(Work Bench Structure) : 개발팀이 수행해야할 작업을 계층적으로 분할한것. 잘게 쪼개져야 핸들링이 편하다. P.96그림 참고.

-스케줄링 : WBS를 기초로 일정 정의. 작업 사이의 의존 관계를 파악하고 CPM방법을 이용한 여유 시간 계산-> 소요자원 할당. (여유시간 = TL-TE) TL=최장경로, TE=최단경로. P.97,98,100 그림 참고.

-비용예측기법 : 노력과 자원과 기간의 관계. D = E/M (기간 = 노력/자원). 비용 예측의 중요한 변수는 투입되는 엔지니어의 인원수와 작업기간이다.

1. 전문가 판단: WBS를 보고 전문가가 비용 측정. 전문가에게 지나치게 의존한다는 단점이 있음.

2. PERT : 낙관적,보통,비관적 일때를 예측해서 가중평균치를 구한다. (낙관적+4\*보통+비관적)/6

3. 알고리즘식 방법 : ex) COCOMO81

-COCOMO81 : 규칙을 기반으로 하는 수학적 공식 사용. 개발 소프트웨어의 유형,팀,프로젝트 프로세스 등 영향주는 요소를 고려한다. 노력=A\*(size)^B \* M. p.104, 105참고. 전문가 판단이나 PERT보다는 근거가 있는 방법이긴하나 프로젝트의 계획단계에서 하기 때문에 굉장히 어렵고 신뢰성이 떨어지며 주관적이라 보장이 어렵다.

-COCOMO2 : COCOMO81의 개선책( 계획단계에선 사이즈 추측이 어렵다는 단점). 소프트웨어 개발 진행도에 따라 3가지 다른 모델을 제시한다.

1. 프로토타입 만들기 : 초기분석단계 -> 사용자요구 수집. 응용점수 계산

2. 초기설계단계 : 설계 -> 기능점수 계산

3. 구조설계이후단계 (모델설계) (프로그램설계) : COCOMO81의 Line size계산 사용.

**-1. 프로토타입 만들기**

추정과정

1. 어플리케이션 구성하는 화면, 보고서, 3세대 언어 컴포넌트 숫자 count
2. 화면과 보고서의 복잡도 수준 결정
3. 화면과 보고서, 3세대 언어 컴포넌트의 개수에 가중치 곱하여 객체점수 계산
4. 화면, 보고서, 3세대 언어 컴포넌트의 개수에 가중치 곱하여 객체점수 계산
5. 재사용률 예측 -> NOP구함
6. PROD 결정
7. PM = NOP / PROD에 대입하여 최종 PM구함

E=b\*S^c \*m(x)

p.107, 108~109 참고.

**-2. 기능점수** : 정확한 라인 수는 예측불가, 각 기능값에 가중값, 입출력,질의, 파일, 인터페이스 개수로 소프트웨어 규모나타냄. 기능 점수 1을 구현하기위한 LOC. 복합 가중값을 이용한 기능점수 산출. 총 라인수 = FP\*원하는 언어의 1점당 LOC. 개발노력 = 총 라인수 / (LOC/MM)

-기능점수 기본 개념 : GFP(총 기능점수)와 PCA(처리 복잡도 보정계수)를 곱한 것. FP = GFP \* PCA. 기능 점수는 구현되는 언어에 관계 없는 매트릭이다. 기능점수 방법은 모든 항목에 일률적인 가중치가 적용되므로 문제가 있을 수 있다. P.112참고.

-기능점수 구하는 법 :

1. 다섯가지 기능 분야에 대한 개수 파악
2. 복잡도 결정
3. 1번,2번 가지고 GFP를 구함
4. 처리 복잡도계산포이용해서 복잡도에 따라 0~5까지 할당
5. PCA구한다. (0.65<=PCA<=1.35)
6. FP = GFP \*PCA로 기능점수구함

p.111, p.112예제 참고.

-국내 기능점수 산정 가이드 : 외부입력, 외부 출력, 내부 논리파일, 외부 인터페이스 파일, 외부 조회, 큰 틀은 COCOMO2의 초기 설계 모델을 따른다.

프로젝트 팀 조직 : 역할과 책임 어디에 있는지, 어떤 통로로 정보가 전달되고 결정되는지 어떻게 갈등을 해소하는지를 정한다.

-구성 : 일회성조직. 소프트웨어 개발 생산성에 큰 영향. 작업의 특성과 팀 구성원 사이의 의사교류.

-역할 나누기 : 프로젝트 관리자, 시스템 운영자, 시스텡ㅁ 분석가, 시스템 개발자, DB엔지니어, QA관리자, 기술지원, HW엔지니어, 웹 개발자, 디자이너

**직능별 조직** : p.117. 서로 다른 부서가 한 프로젝트의 다른 단계에 들어와있음. 팀원은 한 부서에 소속되며 협력도 부서별로 함.

**프로젝트별 조직** : p.118그림 참고

**매트릭스 조직** : 1. 강한 매트릭스 = 프로젝트 일 더 시킴, 2. 약한 매트릭스 = 설계 등의 일을 더 시킴(일을 겸임)

**애자일 조직** : 서로 밀접하게 협력하는 **5~9명**의 팀. 결과와 이슈에 대한 오너쉽을 공유. 대형 프로젝트에는 적합하지 않음.

실행과 모니터링

-프로젝트 실행 : 작업 시작 미팅, 작업 결과 수집

-프로젝트 모니터링 : p.121그림 참고

-모니터링 : 1. 일정 모니터링 = 실제 계획대로 되고 있는지 그림으로 파악가능 (p.122), 2. 어닝 밸류 모니터링 = 예산과 실제 비용, 어닝밸류를 다 비용으로 나타냄(P.123), 3. 번다운 차트 = 남아있는 작업에 초점(p.123) (애자일 프로세스)

리스크 관리

-목적 : 리스크 발생되었을 때 영향을 줄임. P.124그림 참고.

-리스크 파악 : 회의, 문서분석, 리스크 분할구조, 체크리스트, 유추

-리스크 평가 : 영향도에 따라 평가, 우선순위 매김. (확률과 리스크가 발생했을때 미치는 영향이 우선순위 좌우함). 정성적 방법= 확률을 모를때 사용. (risk 매트릭스 사용-> p.125)

-프로젝트 계획서 : p.126~p.127