

3강. 프로젝트 관리

■ 학습개요

소프트웨어 프로젝트는 항상 예산과 일정에 맞춰 진행되어야 하므로 효율적으로 관리되어야 한다. 프로젝트 관리자는 제안서를 작성하고 일정과 비용에 대한 계획을 세우며 프로젝트를 감시하고 통제할 책임을 가진다. 또한 발생할 수 있는 위험을 예측하고 관리하여야 한다. 이 장에서는 프로젝트 관리자의 업무를 학습할 것이다. 소프트웨어의 규모와 비용을 추정하는 방법, 일정 계획을 위한 방법을 알아볼 것이다.

■ 학습목표

1	소프트웨어 프로젝트 관리의 중요성을 이해한다.
2	프로젝트 관리자의 업무를 기술할 수 있다.
3	프로젝트 일정 계획 방법을 사용할 수 있다.
4	비용 산정 모델인 COCOMO를 설명할 수 있다.
5	소프트웨어 프로젝트의 팀 구성 방식을 구별할 수 있다.
6	위험 요인의 분석과 관리 방법을 기술할 수 있다.

■ 주요용어

용어	해설
소프트웨어 프로젝트 관리	프로젝트를 계획하고 계획된 시간과 예산에 맞추어 개발이 진행되는지 감독하는 일
인원-월(PM)	작업 양의 단위로, 1PM은 보통 수준의 1인이 1개월간 수행해야 하는 작업 양
WBS	프로젝트 수행을 위한 개발 업무를 세분화하여 계층구조로 표현한 것
CPM	작업들의 순서를 정하고 임계경로에 기초하여 프로젝트의 일정을 계획하기 위한 방법
COCOMO	프로그램의 규모를 가지고 프로젝트 비용을 산정하는 방법
매트릭스 조직	프로젝트 조직과 기능별 조직을 혼합한 형태의 팀 구성방식으로 개발자는 전문 분야에 속하되 일정 기간 동안 프로젝트에 소속된다.
책임 프로그래머 팀	중앙 집중형 팀 구성 방식으로 책임 프로그래머가 중요한 기술적 판단과 관리적 결정을 한다.
위험 관리	발생 가능한 위험 요인들을 예측하고 영향력을 분석하여 대책을 계획하고 감독하는 것

3.1 프로젝트 관리 개요

프로젝트 관리

- 예산과 일정의 제약으로 관리가 필요
- 프로젝트를 계획하고 프로젝트가 요구되는 표준을 따르는지 확인하고 시간과 예산에 맞추어 개발되는지 감독

소프트웨어 프로젝트 관리의 어려운 점

- 소프트웨어 제품이 보이지 않기 때문에 문서에 의존함
- 소프트웨어 개발 프로세스에 관한 명확한 표준이 없음
- 기술 발전 속도가 빨라 프로젝트 경험을 살리기 어려움

프로젝트 관리자의 업무

- 프로젝트 착수: 제안서 작성
- 프로젝트 계획: 일정, 비용, 자원, 위험 계획
- 프로젝트 실행: 계획에 기초한 프로젝트 감시와 통제
- 프로젝트 종료: 보고서 작성, 프로젝트 평가

프로젝트 계획

- 어느 정도의 비용으로 누구에 의하여 언제까지 행해져야 하는가를 결정
- 발생 가능한 문제점을 분석하고 해결 방안을 모색
- 블록스의 법칙
 - “일정이 늦어진 소프트웨어 프로젝트에 인력을 추가하는 것은 일정을 더욱 늦추는 결과를 낳는다”
 - <- 기존 업무의 이해, 의사소통 경로의 증가, 작업재분할

프로젝트 계획서의 구성

- 개요
- 개발 절차 계획
- 인원, 예산 및 일정 계획
- 문서화 계획
- 하드웨어와 소프트웨어 자원 계획
- 위험 관리 계획

3.2 소프트웨어 일정 계획

일정 계획을 위한 작업

- 작업의 분할
 - 전체 작업을 관리 가능하고 측정 가능한 소작업들로 분할
 - 작업 분해 구조(WBS)
- 작업의 명세화
 - 소작업들에 대해 일의 양, 필요한 산출물(이정표)과 컴퓨터 자원 등을 결정
 - 작업의 양을 인원-월(PM)로 표시함, 1PM은 중급 수준 개발자의 한 달간 작업량
- 작업 진행 순서의 정의
 - 작업들의 선행 관계를 분석하여 순서를 정함
 - PERT/CPM
- 인력 배정
 - 작업의 양과 특성에 맞도록 개발자를 배정
- 작업 비용의 산정
 - 작업의 양과 인력에 따른 비용을 산정
- 개발 일정의 수립
 - 작업별로 시작 시점과 종료 시점을 설정
 - CPM, 간트(Gantt) 차트

이정표(milestone)와 결과물(deliverable)

- 이정표
 - 소프트웨어 프로세스에서 개별 활동의 종료 시점 또는 주요 단계의 종료 시점에서 얻어지는 중간 산출물
 - 프로젝트 진행 상황의 파악, 비용과 일정의 수정을 위해 필요함
- 결과물
 - 프로젝트 주요 단계의 종료 시점에서 고객에게 전달되는 산출물

WBS(Work Breakdown Structure)

- 작업 분할 구조
- 프로젝트 수행을 위해 개발 업무를 분할하여 계층 구조로 표현
- 최하위 수준의 작업을 작업 패키지(work package)라고 함
 - 정량적으로 측정 가능한 입력물과 출력물을 가짐
- 프로젝트 계획과 관리를 위한 기초 자료

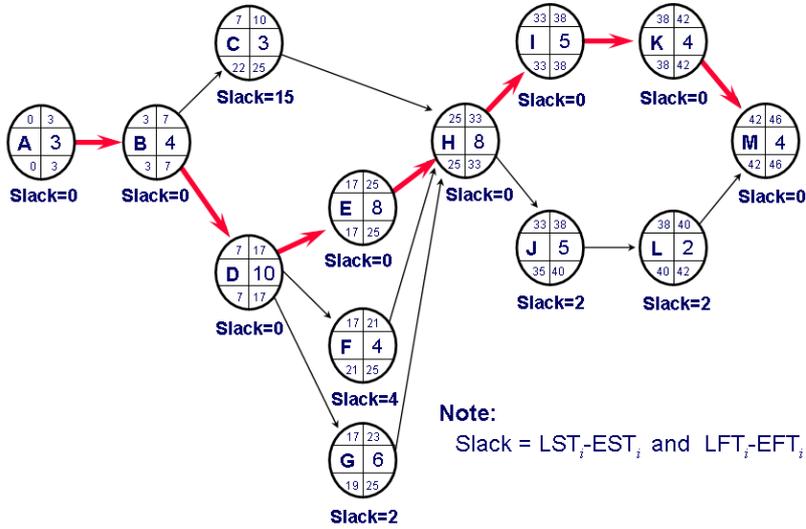
PERT

- 작업들의 선후 관계를 표현
- 작업 패키지들의 순서에 관한 정보로 사이클이 없는 방향 그래프

CPM

- 임계 경로 방법
- 임계 경로는 시작에서 종료 작업까지의 경로 중 가장 긴 경로
 - 임계 경로상의 작업들은 프로젝트의 일정 준수를 위해 지연이 허용되지 않는 작업

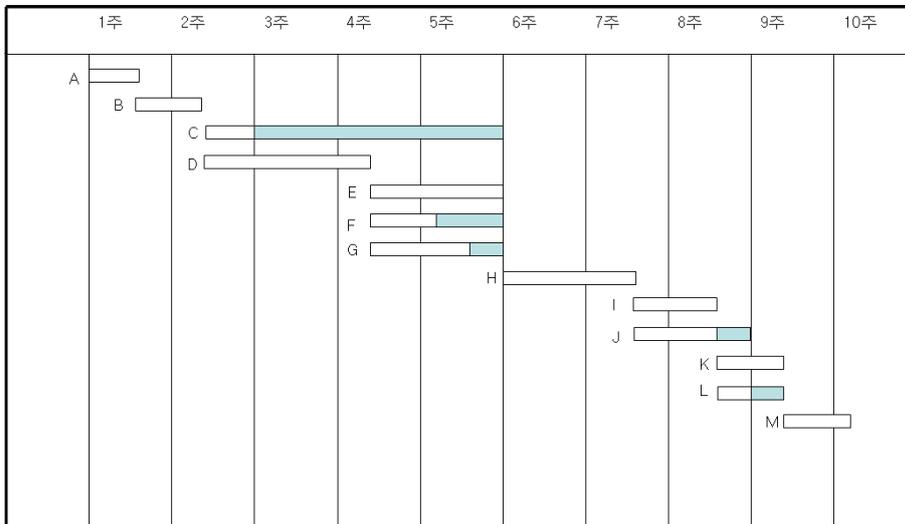
임계 경로상에 있지 않은 작업들은 여유 시간을 가짐



[그림 3-6] CPM 네트워크와 경계 시간 및 임계 경로

간트(Gantt) 차트

- 막대 모양으로 프로젝트 작업들의 순차 또는 병행 순서를 보여주는 차트 상단에 시간축을 표시, 작업별로 막대를 표시
막대는 작업 시간에 맞추어지며 길이는 소요 시간을 의미
- 인력 배정 등의 자원 활용 계획에도 사용됨



[그림 3-7] 간트 차트

3.3 소프트웨어 규모의 산정

소프트웨어 프로젝트 산정

- 프로젝트 완성을 위해 소요되는 여러 요소를 추정하는 일
소프트웨어 규모, 개발 비용, 기간 등

- 프로젝트 계획 작업이며 프로젝트 실행과 관리에 기초가 됨

라인 수(LOC)

- 간단하며 비용 산정 방법과의 연결이 용이하고 가시적임
- 계획단계에서 산정하기 어렵고 프로그래밍 언어에 따라 다름
- 과거의 경험, 전문가의 판단, 구성 요소별 산출한 후 합산

기능 점수(FP)

- 기능의 규모를 측정하기 위한 단위
- 프로그램의 기능에 초점을 맞춘 논리적 규모 척도
 - 기능적 사용자 요구사항을 양으로 표시
- 구현 기술이나 구현 언어와 무관
- 사무 정보 시스템의 규모 산정에 적합함
- 조정 기능 점수(AFP)는 미조정 기능 점수(UFP)와 조정 계수(VAF)의 곱

미조정 기능 점수(UFP)

- 자료의 규모로부터 기능을 측정
- 프로그램에서 표현되거나 사용된 데이터의 총량을 계량화
- 데이터 기능(논리적 내부 파일, 외부 인터페이스 파일)과 트랜잭션 기능(외부 출력, 외부 입력, 외부 조회)을 측정
 - 각각의 개수에 유형의 복잡도에 따른 가중치를 곱하고 모두 합함

조정 계수(VAF)

- 기술적 복잡도를 반영하기 위해 14개 복잡도 항목의 영향도를 계산하여 모두 합함
- $VAF = 0.65 + 0.01 * TDI$

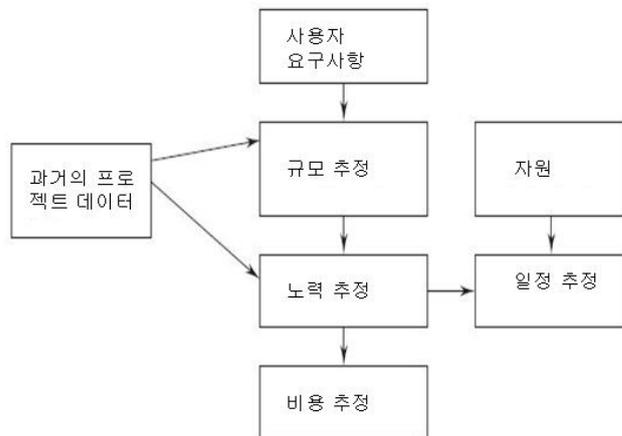
조정 기능 점수(AFP)

- $AFP = UFP * VAF$
- 프로그래밍 언어별로 기능 점수 1점을 구현하기 위해 필요한 라인 수가 존재
 - 기능 점수로부터 라인 수를 계산할 수 있음
 - 초기 단계에서 라인 수 추정에 효과적
- 프로그래머의 평균 생산성(FP/PM)이 알려져 있다면 전체 PM을 계산할 수 있음

3.4 소프트웨어 개발 비용 산정

소프트웨어 프로젝트 산정

- 규모 추정, 개발 노력(비용) 추정, 일정 계획
- 노력 추정에 과거 데이터와 프로젝트의 특성을 고려
- 일정 계획에는 투입 되는 인력을 고려
- 추정의 정확성은 과거 프로젝트 데이터의 정확성, 제공되는 입력의 정확성, 개발 조직에서 프로세스의 성숙도에 좌우됨



[그림 3-8] 프로젝트 규모, 노력, 비용 및 일정의 추정 과정

비용 산정 방법의 분류

- 판단에 의한 방법
 - 전문가의 판단, 델파이 방법, 작업 분해에 의한 방법
- 모델을 이용한 방법
 - 알고리즘 모델, 유추에 의한 산정

COCOMO(ConstrucTive COst MOdel)

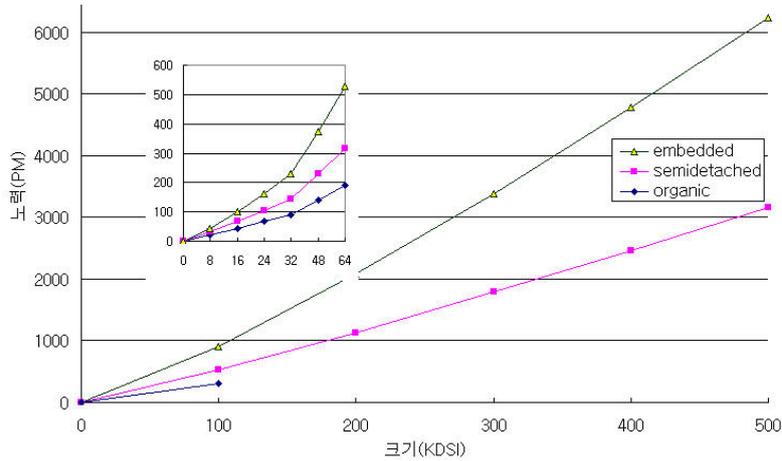
- 가장 잘 알려진 소프트웨어 비용 산정 모델
- 프로젝트 유형을 3가지로 구분하고 분석 정도에 따른 3가지 모델을 제시
- 기본 COCOMO
 - 라인 수로 노력을 계산
- 중급 COCOMO
 - 비용 승수를 고려
- 상세 COCOMO
 - 서브시스템이나 모듈별로 비용을 계산
 - 분석, 설계 등의 개발 단계별로 비용을 계산
- 프로젝트 유형
 - 기본형: 소규모 프로젝트, 경험있는 개발자, 까다롭지 않은 요구사항
 - 중간형: 중규모 프로젝트, 중간 정도의 경험, 요구사항의 혼재
 - 내장형: 대규모 프로젝트, 엄격한 제약조건

기본 COCOMO

- 대략적으로 개발 노력은 소프트웨어 규모에 선형적으로 비례

<표 3-5> 기본 COCOMO 공식

	총 노력(PM)	총 개발 기간(개월)
기본형	$Effort=(2.4 \cdot KLOC^{1.05})$	$T_{dev}=2.5 \cdot (Effort)^{0.38}$
중간형	$Effort=(3.0 \cdot KLOC^{1.12})$	$T_{dev}=2.5 \cdot (Effort)^{0.35}$
내장형	$Effort=(3.6 \cdot KLOC^{1.20})$	$T_{dev}=2.5 \cdot (Effort)^{0.32}$



[그림 3-9] 프로젝트 크기에 따른 노력 (프로젝트 유형별)
-예

$$\text{총 노력} = 2.4 \cdot 32^{1.05} = 91 \text{ PM}$$

$$\text{총 개발 기간} = 2.5 \cdot (91)^{0.38} = 14 \text{ 개월}$$

$$\text{평균 인력} = 91/14 = 6.5 \text{ 명}$$

$$\text{생산성} = 32000/91 = 352 \text{ LOC/PM}$$

중급 COCOMO

- 15개의 비용 승수를 곱하여 노력 조정 계수(EAF)를 계산
 각 비용 승수는 6개의 등급으로 나뉜
- 총 노력을 계산할 때 EAF를 곱함

<표 3-6> COCOMO 비용 승수

승수	비용 승수값					
	매우낮음	낮음	보통	높음	매우높음	극히높음
제품의 속성						
요구되는 S/W 신뢰도	0.75	0.88	1.00	1.15	1.40	
데이터베이스 크기		0.94	1.00	1.08	1.16	
제품의 복잡도	0.70	0.85	1.00	1.15	1.30	1.65
컴퓨터 속성						
실행 시간의 제약			1.00	1.11	1.30	1.66
주기억 장치의 제약			1.00	1.06	1.21	1.56
컴퓨터와 S/W의 안정성		0.87	1.00	1.15	1.30	
요구되는 응답 시간		0.87	1.00	1.07	1.15	
개발 요원의 속성						
분석가의 능력	1.46	1.19	1.00	0.86	0.71	
응용에 대한 경험	1.29	1.13	1.00	0.91	0.82	
엔지니어의 능력	1.46	1.17	1.00	0.86	0.70	
컴퓨터와 S/W의 친숙도	1.21	1.10	1.00	0.90		
프로그래밍 언어 경험	1.14	1.07	1.00	0.95		
프로젝트 속성						
소프트웨어 공학의 활용	1.24	1.10	1.00	0.91	0.82	
도구의 사용	1.24	1.10	1.00	0.91	0.83	
요구되는 개발 일정	1.23	1.08	1.00	1.10	1.10	
예: 산출된 노력 조정 계수 ≃ 1.17						

<표 3-7> 중급 COCOMO 공식

	총 노력(PM)	전체 개발 시간(개월)
기본형	$Effort=(3.2 \cdot KLOC^{1.05}) \cdot EAF$	$T_{dev}=2.5 \cdot (Effort)^{0.38}$
중간형	$Effort=(3.0 \cdot KLOC^{1.12}) \cdot EAF$	$T_{dev}=2.5 \cdot (Effort)^{0.35}$
내장형	$Effort=(2.8 \cdot KLOC^{1.20}) \cdot EAF$	$T_{dev}=2.5 \cdot (Effort)^{0.32}$

- 예

$$\text{총 노력} = 2.8 \cdot 4^{1.20} \cdot 1.17 = 17 \text{ PM}$$

$$\text{총 개발 기간} = 2.5 \cdot 17^{0.32} = 6 \text{ 개월}$$

$$\text{비용} = 17\text{PM} \cdot 5,000,000\text{원/PM} = 850,000,000\text{원}$$

소프트웨어의 수정을 위한 노력

- 설계, 코드, 통합과 테스트 부분에서 수정이 필요한 비율을 구하여 수정 조정 계수(AAF)를 계산
- $AAF = 0.4 \cdot (\text{설계 수정 비율}) + 0.3 \cdot (\text{코드 수정 비율}) + 0.3 \cdot (\text{통합과 테스트 수정 비율})$
- AAF를 이용하여 수정이 요구되는 LOC를 계산한 후 공식에 대입

3.5 팀 구성 방식

매트릭스 조직

- 프로젝트 조직과 기능별 조직의 장점을 조합한 형태
- 개발자가 전문 분야에 속하되 일정 기간 프로젝트에 소속되는 형태

- 팀 구성원들 간에 정보와 경험을 공유할 수 있으나 기능 부서 관리자와 프로젝트 관리자 양쪽의 지배를 받음



[그림 3-10] 매트릭스 조직

의사 결정 방법에 따른 팀 구성

- 비이기적인 팀
 - 분산형 팀 구성 방식
 - 구성원 전체가 의사 결정에 참여
 - 작업에 대한 만족도가 높으나 의사 결정이 늦고 책임 소재가 모호함
- 책임 프로그래머 팀(chief programmer team)
 - 중앙 집중형 팀 구성 방식
 - 책임 프로그래머가 중요한 기술적 판단과 관리적 결정을 함
 - 의사소통 경로가 감소하나 책임 프로그래머의 능력에 크게 의존함
- 계층형 팀 구성
 - 중앙 집중형과 분산형 팀 구성 방법을 혼합

3.6 위험 분석과 관리

위험

- 불확실성으로 인한 잠재적인 문제
- 비용, 품질 및 일정에 좋지 않은 영향을 줄 수 있는 부정적 요인
- 위험 관리
 - 발생 가능한 위험 요인들을 예측하고 영향력을 분석하여 대책을 계획하고 프로젝트를 실행하는 것

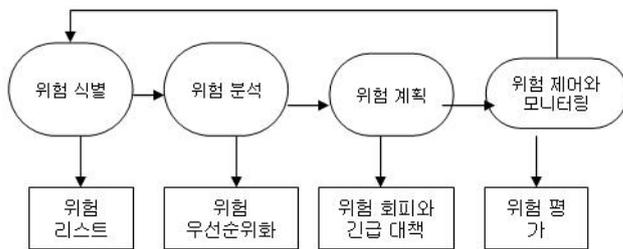
위험의 분류

- 제품 위험
 - 제품의 품질이나 성능에 영향을 주는 위험
 - 불안정한 요구사항이나 성능이 낮은 도구 등
- 조직 위험
 - 조직의 비즈니스에 영향을 주는 위험
 - 기술의 변화나 경쟁사 제품의 출시 등
- 프로젝트 위험
 - 프로젝트 일정이나 자원 활용에 영향을 주는 위험

미흡한 조직의 지원, 중요 프로젝트 요원의 이직 등

위험 관리 프로세스

- 위험 식별
발생 가능한 위험 요인을 나열
- 위험 분석
위험 요인별로 발생 가능성과 충격의 심각성을 평가
- 위험 계획
회피 전략: 발생 가능성을 줄이는 것
최소화 전략: 위험 발생시 충격을 줄이는 것
긴급 대책: 최악의 상화에 대비하는 것
- 위험 제어와 모니터링



[그림 3-14] 위험 관리 프로세스

메 뉴	연습문제
----------------	-------------

1. 브룩스의 법칙이 의미하는 것이 아닌 것은?

- ① 투입 인력의 증가로 의사소통 경로가 크게 증가한다
- ② 1인이 12개월 할 일을 12인이 1개월에 할 수 없다
- ③ 가능한 초기에 많은 인력을 투입하는 것이 좋다
- ④ 새로 투입된 인력은 진행된 작업을 먼저 이해해야 한다

<정답> ③

<해설> 브룩스의 법칙은 “지연되고 있는 프로젝트에 더 많은 프로그래머를 투입할 경우 오히려 더 지연될 수 있다”는 것이다.

2. 소프트웨어 프로젝트의 소요 기간과 비용을 추정하는 방법에 관한 설명으로 옳지 않은 것은?

- ① COCOMO는 먼저 완성될 시스템의 규모를 추정해야 한다
- ② 기능 점수 방법은 구현에 이용되는 프로그래밍 언어에 독립적이다

- ③ 기능 점수 방법으로 비용을 산정하려면 프로그래머의 생산성(FP/PM)을 알아야 한다
- ④ 인원-월(PM)은 프로젝트 수행 기간을 나타내기 위한 단위이다.

<정답> ④

<해설> 1PM은 중간 수준 개발자의 1개월 작업량을 의미하는 것으로 노력을 추정할 때 사용되는 단위이다.

3. 프로젝트 일정 계획과 관련이 없는 것은?

- ① 작업 분할 구조(WBS)
- ② 간트(Gantt) 차트
- ③ 임계 경로 방법(CPM)
- ④ 자료 흐름도(DFD)

<정답> ④

<해설> 일정 계획을 위해 먼저 작업 분할 구조가 만들어져야 한다. 간트 차트는 일정 계획을 표현하기 위한 차트이다.

4. 소프트웨어에 요구되는 신뢰도와 개발 비용과의 관계는 무엇인가?

- ① 신뢰도와 개발 비용은 무관하다
- ② 신뢰도와 개발 비용은 반비례한다
- ③ 신뢰도가 증가하면 개발 비용은 선형적으로 비례한다
- ④ 신뢰도가 높아질수록 개발 비용은 크게 증가한다

<정답> ④

<해설> 일반적으로 높은 신뢰도가 요구되는 경우 정형적인 요구사항 명세나 엄격한 테스트 작업이 필요하여 개발 비용은 매우 크게 증가하게 된다.

5. 소프트웨어 개발 팀 구성에 관한 설명으로 적당한 것은?

- ① 중앙 집중식 팀 구성은 비교적 소규모 프로젝트에 적합하다
- ② 분산형 팀의 의사소통 경로는 계층적이다
- ③ 분산형 팀에서 1명의 작업자가 추가되면 의사소통 경로는 1만큼 증가한다
- ④ 매트릭스 조직은 부서를 기능별로 전문화하지 않는다

<정답> ①

<해설> 분산형 팀 구성에서 작업자 1인이 추가되면 $(n-1)$ 의 의사 소통 경로가 추가된다. 매트릭스 조직은 기능 부서별로 조직화되어 일정 기간 프로젝트에 소속되는 형태이다.

1. 소프트웨어 프로젝트 관리자의 주요 업무는 무엇인가?

프로젝트 제안서 작성, 일정/비용/자원/위험 계획, 계획에 기초한 프로젝트의 감시와 통제, 최종 보고서의 작성과 프로젝트 평가

2. WBS는 무엇인가?

작업 분할 구조를 의미하는 것으로 프로젝트 수행을 위한 개발 업무를 계층적으로 보여주는 것

3. CPM에서 임계 경로란 무엇인가?

프로젝트 일정을 계획하기 위해 소작업들의 선행 관계를 고려하여 방향 그래프로 표현했을 때 시작에서 종료까지의 가장 긴 경로이다. 임계 경로 상의 작업이 지연되면 전체 일정이 지연된다.

4. 조정 계수(VAF)를 사용하여 기능 점수를 계산하는 방법을 설명하라.

열 네가지 항목에 대해 기술적 복잡도에 미치는 영향도(0~5)를 판단하고 모두 합하여 총 영향도(TDI)를 계산한 후 이것을 미조정 기능 점수(UFP)에 반영한다.

$$AFP = UFP * (0.65 + 0.01 * TDI)$$

5. 세 가지 COCOMO 모델 중 중급 COCOMO를 설명하라.

소프트웨어 개발에 영향을 미치는 15개의 비용 승수를 고려한 것이다. 각 비용 승수의 등급(매우 낮음~극히 높음)을 정한 후 모두 곱하여 노력 조정 계수(EAF)를 구한 다음 총 노력을 계산할 때 반영한다.

6. 팀 구성 방식 중 책임 프로그래머 팀의 장단점을 설명하라.

중앙 집중형 팀 구성 방식으로 의사 소통 경로가 감소하나 책임 프로그래머의 기술적/관리적 능력에 크게 의존한다.