

3. 프로젝트 관리

학습목표

- ❖ 관리 활동
- ❖ 프로젝트 계획 수립
- ❖ 프로젝트 일정 관리
- ❖ 소프트웨어 비용산정

흔한 단어, XX 프로젝트?



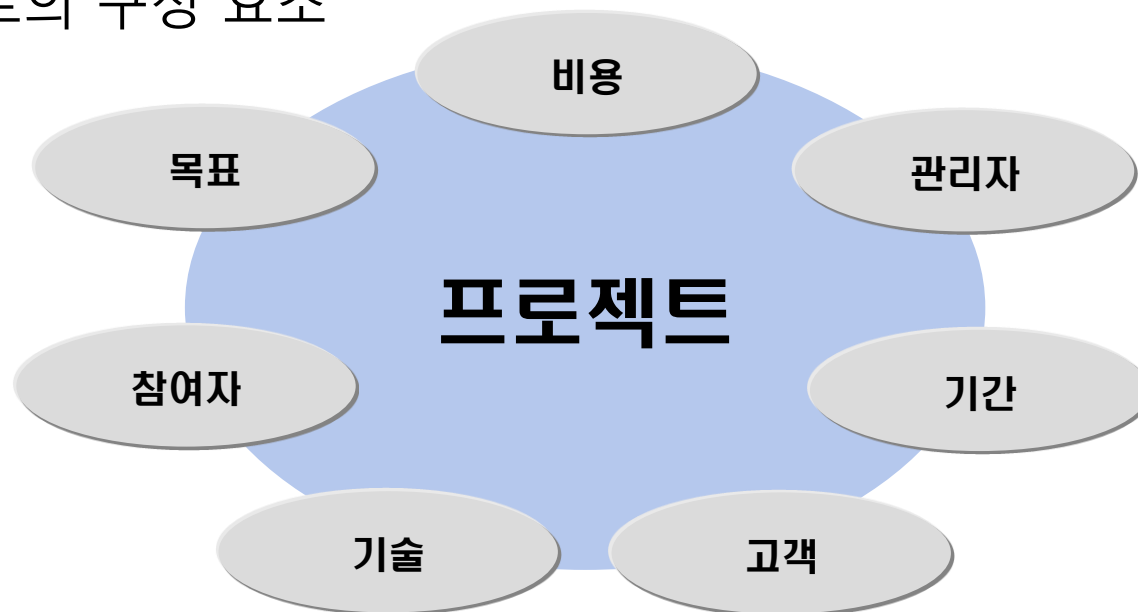
프로젝트란?

❖ 정의

- 프로젝트는 **유일한** 제품이나 서비스를 만들기 위해 수행되어야 할 **일시적인** 행동
- 같은 일을 반복하는 "일상생활"과 구분됨

출처: PMBOK

❖ 프로젝트의 구성 요소



소프트웨어 프로젝트

❖ 소프트웨어 개발의 시작

- 조직이 요구사항에 맞는 프로그램을 조달(Acquire)할 필요가 생겼을 때
 - 시중에 나와있는 프로그램을 구입
 - 소프트웨어 개발 전문 업체에 의뢰
 - 직접 개발

❖ 소프트웨어 개발

- 발주자(고객)가 요구사항을 주면 수주자(개발자)가 요구사항에 맞는 프로그램을 개발



소프트웨어 프로젝트 프로세스(1/2)

❖ 소프트웨어 제품 구상

- 발주자
 - 원하는 소프트웨어 제품의 구상 및 그 가치를 검증
 - 제품의 투자 대비 효과를 예측하고, 사업에 미치는 영향을 파악
 - 원하는 제품의 기능상의 요구사항, 성능 요구사항들을 정의

❖ 소프트웨어 제안 요청서(RFP: Request for Proposal) 배포

- 제품을 자체 개발하지 않는 경우, 개발 회사들에게 제안 요청서를 발송

❖ 제안서 제출

- 개발 회사들은 발주자에게 제안서 제출

일반적인 제안 요청서 양식

목 차

- I. 프로젝트 개요
 - 1. 프로젝트 명
 - 2. 프로젝트 목적
 - 3. 프로젝트 결과물
 - 4. 프로젝트 내용
 - 5. 특이사항
 - 6. 기간
 - 7. 비용
 - 8. 추진일정
 - ...
- II. 별지서식

소프트웨어 프로젝트 프로세스(2/2)

❖ 제안서 심사

- 이미 정해진 기준에 따라 심사하여 수주자 선정

❖ 계약서 작성

- 수주자가 선정되면 발주자와 수주자 사이에 계약 체결

❖ 프로젝트 시작 및 수행

- 계약이 완료 후 수주자는 프로젝트 시작
- 마일스톤 별로 또는 발주자의 참여 필요 시 회의를 갖고 요구사항의 변경 등 중요한 사항 협의

❖ 프로젝트 종료 및 제품 인도

- 소프트웨어 개발 완료 후 발주자의 인수 테스트를 거쳐 제품이 인도됨

일반적인 계약서 양식

소프트웨어 개발 계약서

한국 발주사(이하 “갑”이라 함)와 개발 코리아(이하 “을”이라 함)는 제 2조에 명시한 “소프트웨어 개발”의 관련 업무 대하여 다음과 같이 계약을 체결한다

-다 음-

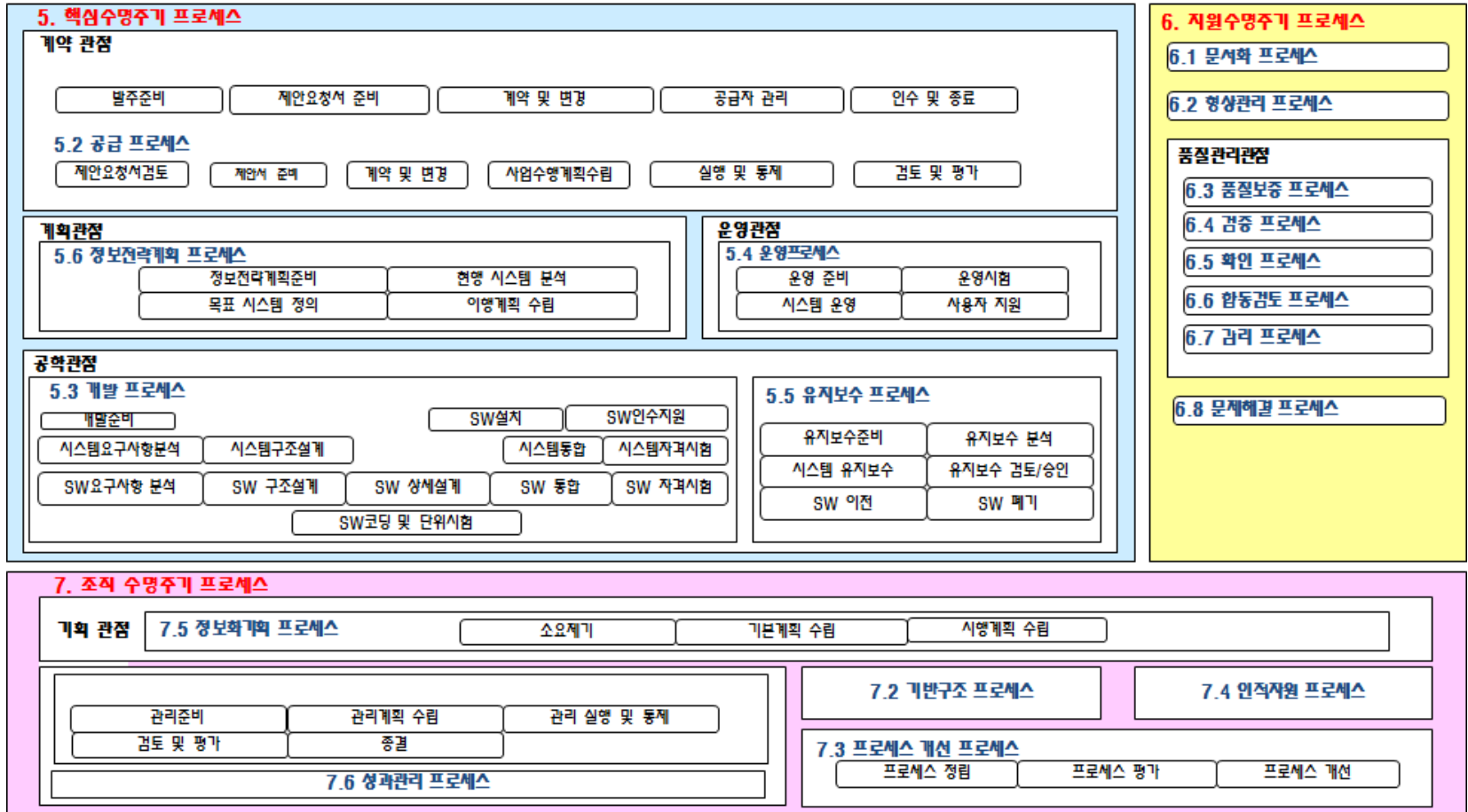
제 1조 계약의 목적
제 2조 계약 내용
제 3조 협조 사항
제 4조 계약 기간
제 5조 계약 금액
제 6조 사용 및 저작권한
제 7조 계약의 해지 및 통보
제 8조 비밀 유지의 의무와 손해배상
제 9조 기타

소프트웨어 프로젝트 관리

❖ 소프트웨어 공학의 특징

- 소프트웨어 제품의 비가시성 및 무형성
 - 소프트웨어 제품은 눈에 보이지도 않고 형체가 없어서 만질 수도 없다.
 - 소프트웨어 프로젝트 관리자는 프로젝트 진척 사항을 볼 수 없다.
 - 프로젝트 진척 사항을 간접적으로 확인할 수 있는 문서화 작업에 의존한다.
- 표준 프로세스 부재
 - 수년간 소프트웨어 프로세스는 진화해왔지만, 다른 공학 분야와 같은 표준 프로세스가 정립된 것은 아니다.
 - ISO 12207은 소프트웨어 개발 단계에서 따라야 할 절차를 정의한 표준 프로세스가 아니라 소프트웨어 프로세스의 각 단계에서 수행하는 표준적인 활동들을 정의한 것임.
- 대규모 소프트웨어 프로젝트의 유일성
 - 대규모 프로젝트의 경우 각 프로젝트 별로 개성이 뚜렷하기 때문에 앞으로 발생할 문제를 예측하기가 어렵다는 의미이다.
 - 기술적으로도 컴퓨터와 통신 분야에서 급격한 발전이 이루어져서 과거의 기술을 전제로 하는 과거 프로젝트의 경험에서 얻어진 지식이 새로운 프로젝트에 활용되지 않는 경우도 많다.

ISO 12207



관리 활동

❖ 대표적인 관리 활동

- 제안서 작성
- 프로젝트 비용 산정
- 프로젝트 계획 및 일정관리
- 프로젝트 감독 및 확인
- 인력 선발 및 평가
- 보고서 작성 및 프레젠테이션

프로젝트 계획 수립

```
프로젝트 제약 조건 설정
프로젝트 인자의 초기 추정
프로젝트 이정표와 산출물 정의
while 프로젝트가 완료되지 않거나 취소 loop
    프로젝트 일정 결정
    일정에 따라 활동 시작
    (잠시) 대기
    프로젝트 진척사항 검토
    프로젝트 인자 추정치 조정
    프로젝트 일정 갱신
    프로젝트 제약 조건과 산출물 재협상
    if (문제가 발생) then
        기술적 검토 및 가능한 수정 시작
    end if
end loop
```

[그림 3-2] 소프트웨어 계획 프로세스

이정표와 산출물

- ❖ 프로젝트의 활동은 진척 사항을 판단하기 위해 유형의 산출물을 생성하도록 구성되어야 함.
- ❖ **이정표 (Milestones)** 는 프로세스 활동의 종료점.
- ❖ **산출물 (Deliverables)** 은 고객에게 전달되는 프로젝트 결과.
- ❖ 폭포수 프로세스는 진척 사항에 관한 이정표의 직접적인 정의를 가능하게 함.

프로젝트 계획서의 구성

- ① 개요
- ② 프로젝트 조직
- ③ 위험 분석
- ④ 하드웨어 및 소프트웨어 자원 요구사항
- ⑤ 업무 분할
- ⑥ 프로젝트 일정 계획
- ⑦ 감독 및 보고체계

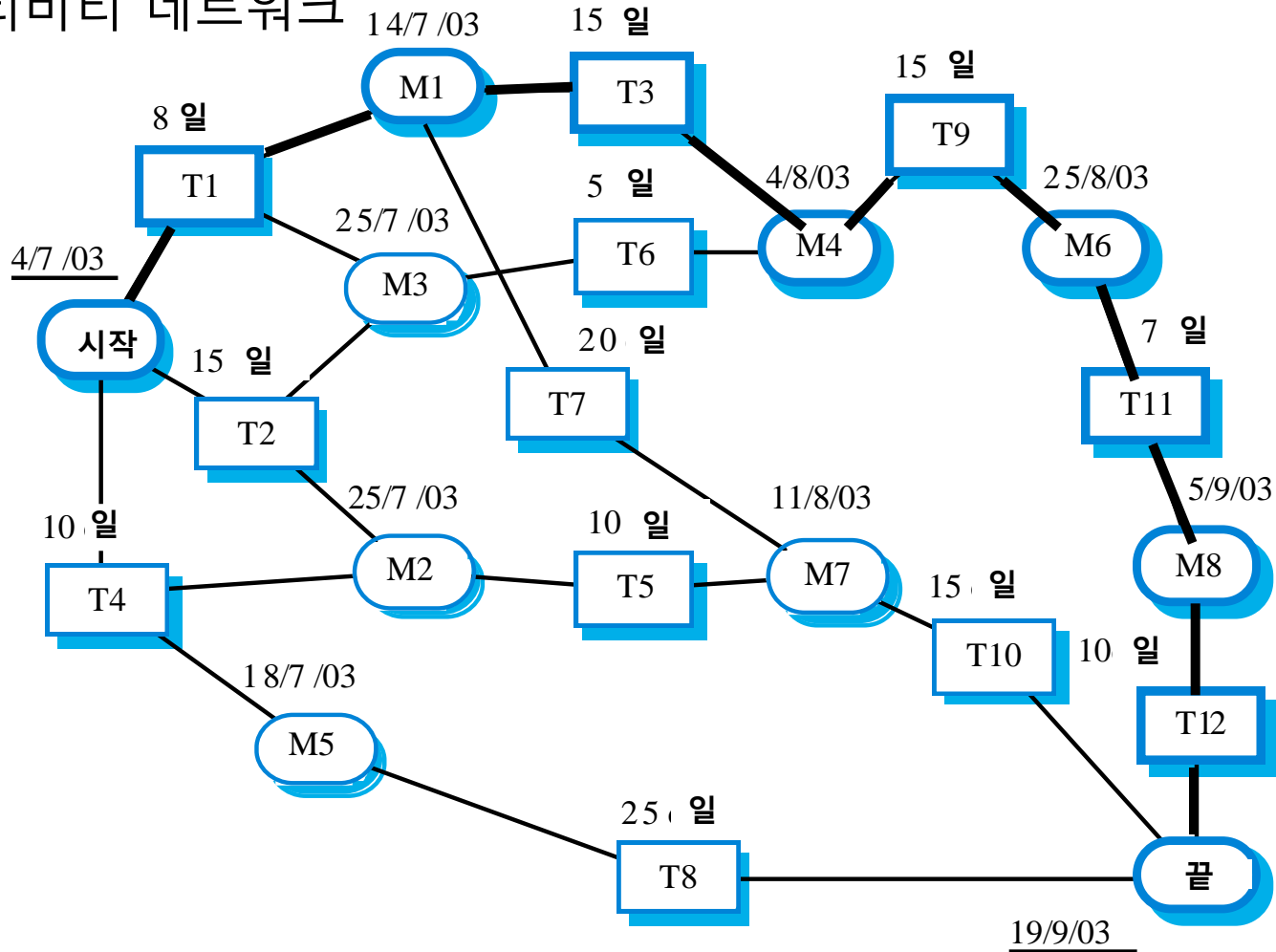
프로젝트 일정 관리 (1/4)

❖ 작업 기간과 종속성

작업	기간(일)	종속성
T1	8	
T2	15	
T3	15	T1 (M1)
T4	10	
T5	10	T2, T4 (M2)
T6	5	T1, T2 (M3)
T7	20	T1 (M1)
T8	25	T4 (M5)
T9	15	T3, T6 (M4)
T10	15	T5, T7 (M7)
T11	7	T9 (M6)
T12	10	T11 (M8)

프로젝트 일정 관리 (2/4)

❖ 액티비티 네트워크

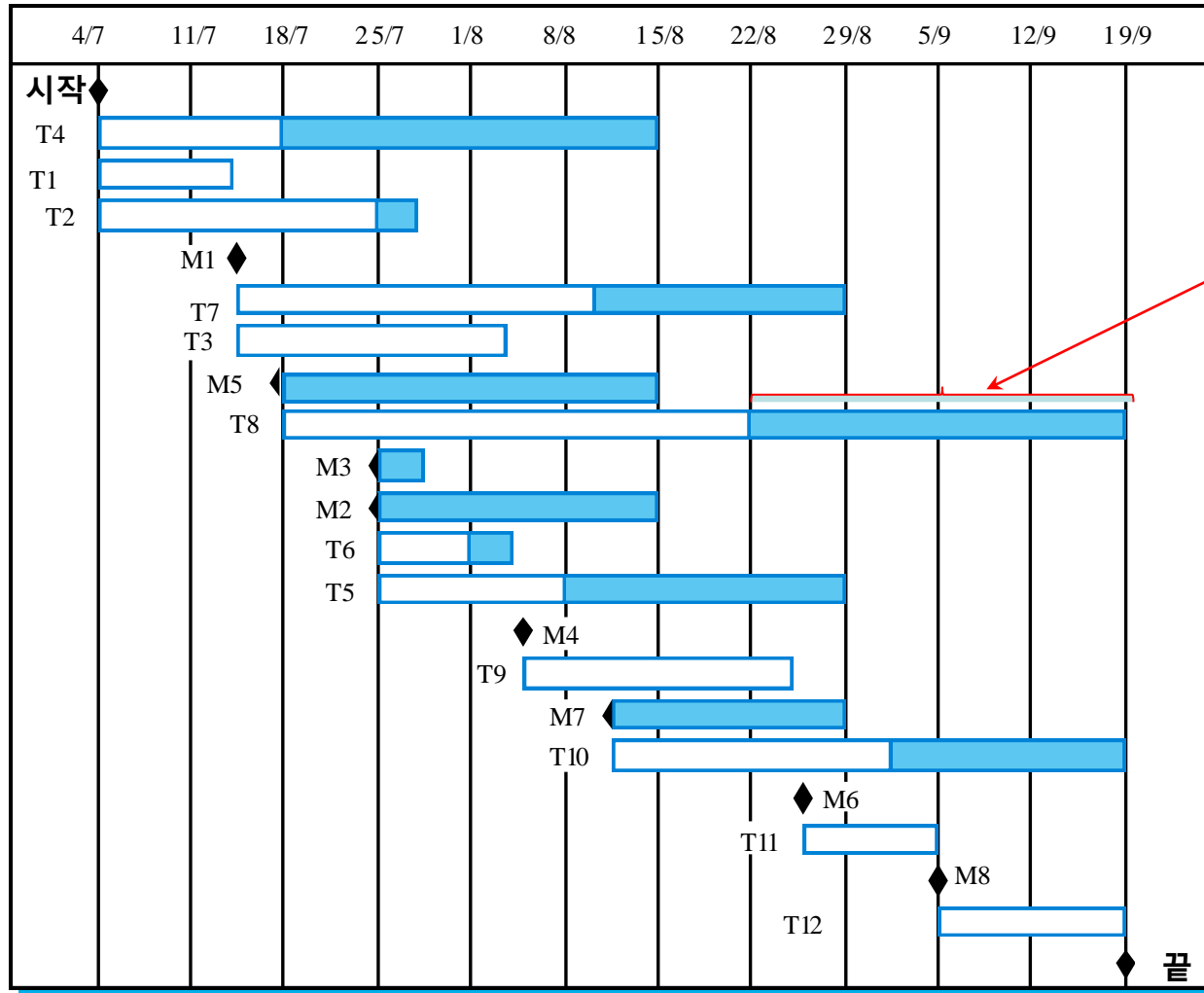


임계경로(Critical Path): 액티비티 네트워크의 가장 긴 경로

- T8은 임계경로 상의 작업이 아니므로 4주 지연되어도 전체 일정에 무영향

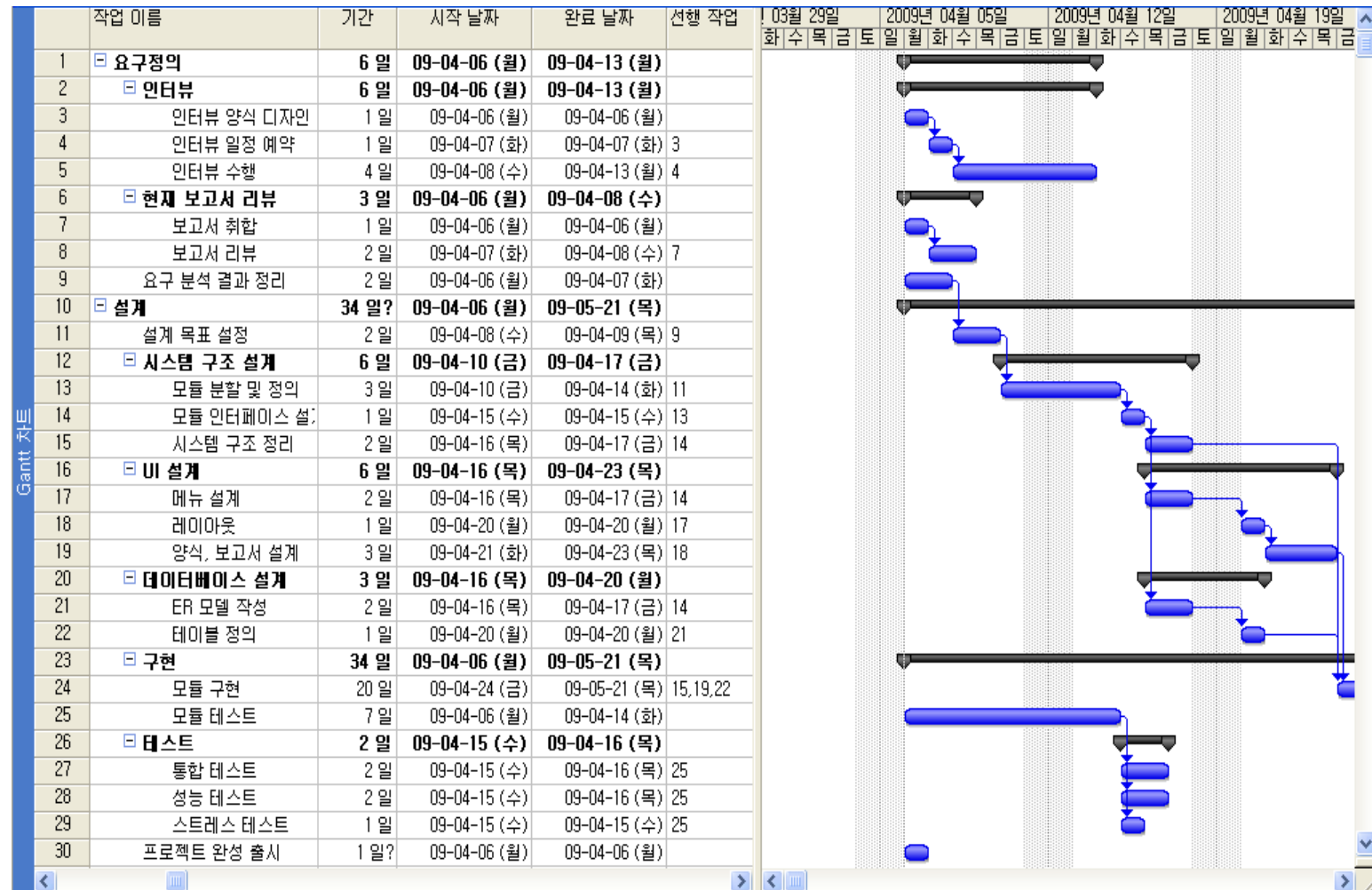
프로젝트 일정 관리 (3/4)

❖ 바차트 (간트 차트)



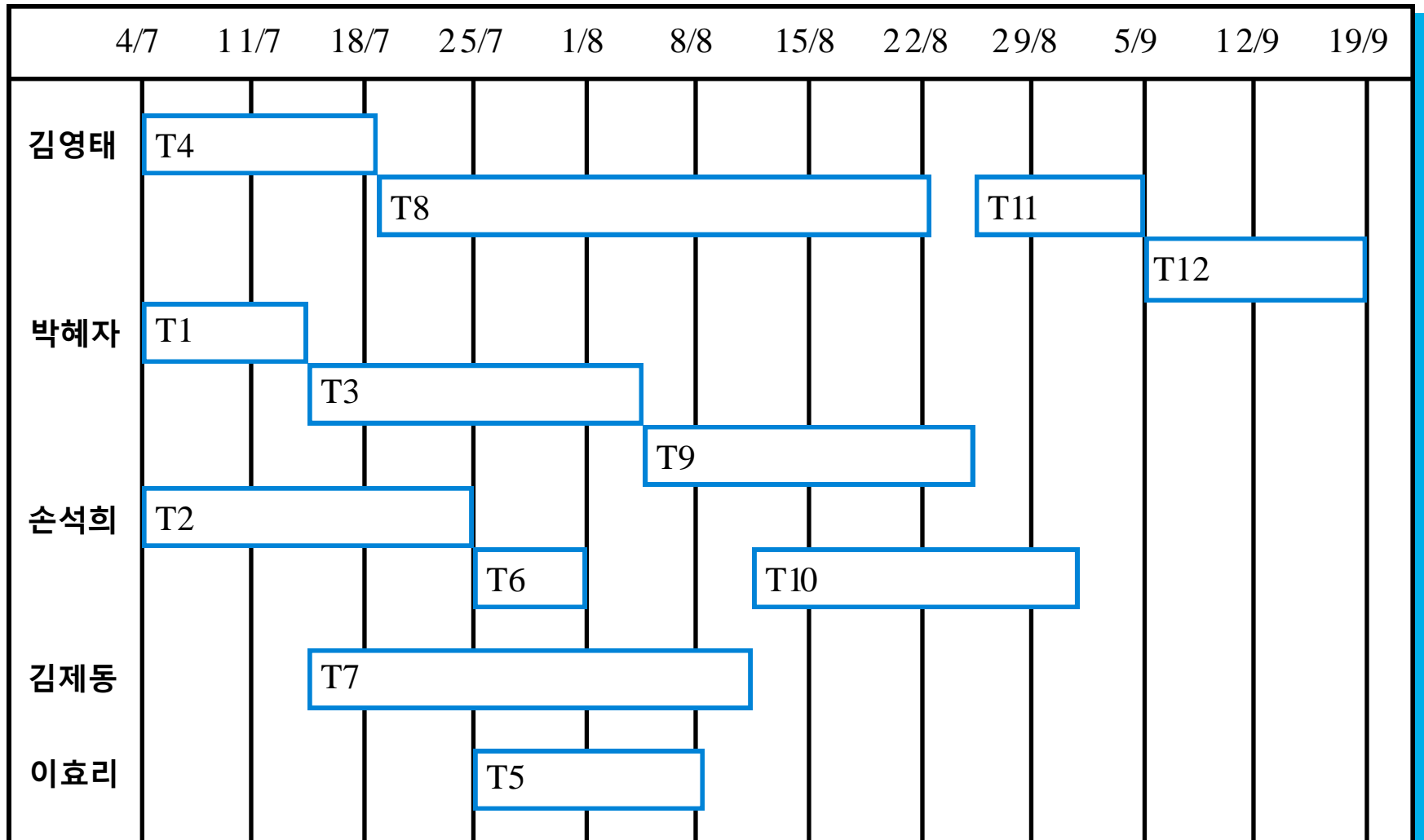
여유 시간
(slack time)

MS Project를 이용한 간트 차트 작성



프로젝트 일정 관리 (4/4)

❖ 인력 배정



소프트웨어 비용산정

❖ 생산성

- LOC/PM
- FP/PM

❖ LOC의 문제점

- 생산성 모순
- 대안으로 기능점수(FP, Function Point) 도입

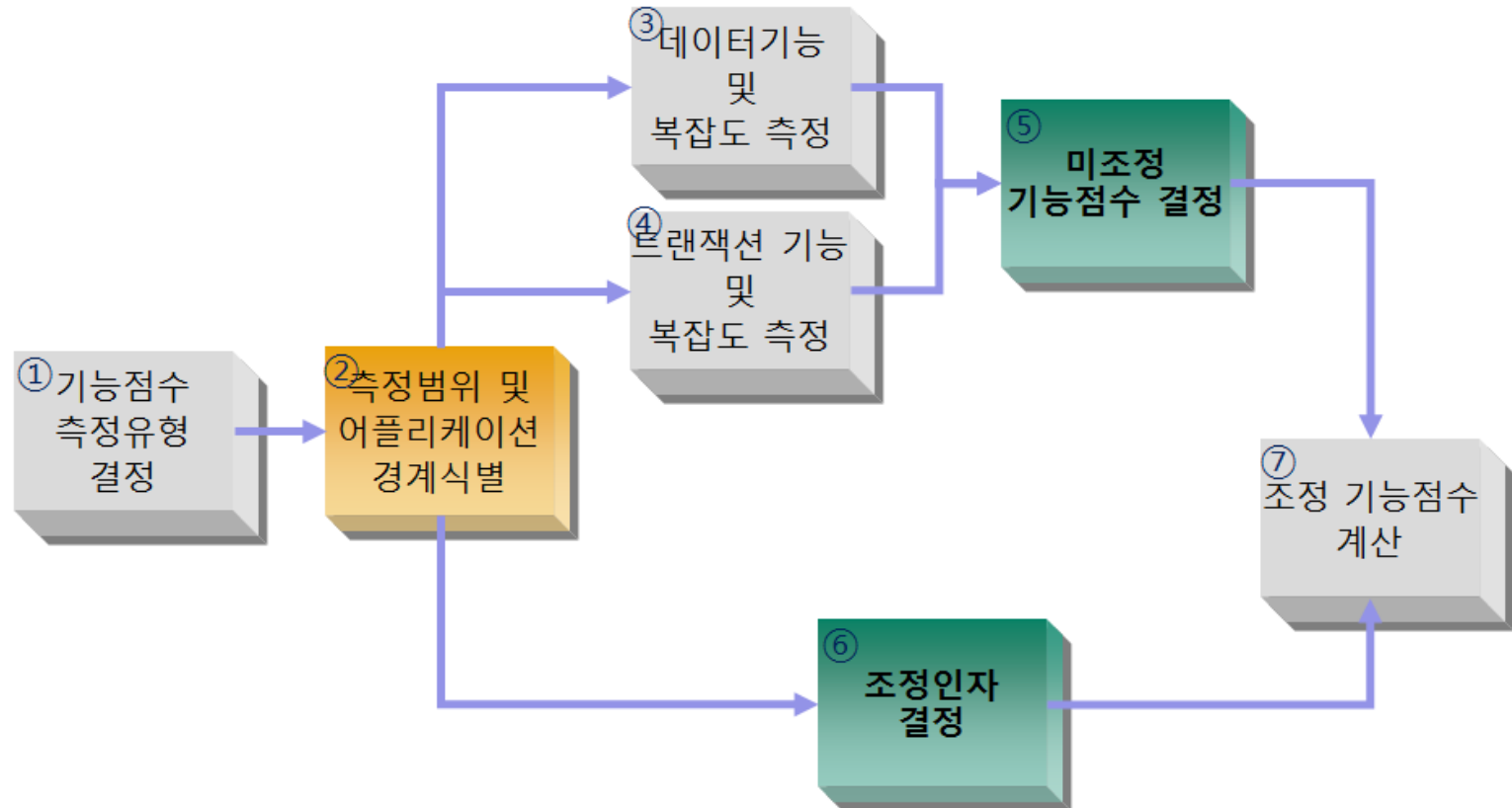
생산성 모순

- ❖ 언어의 수준이 낮을수록, 프로그래머의 생산성은 더욱 높아짐
 - 동일한 기능을 구현하기 위해 고급 언어보다 저급 언어가 더 많은 코드가 필요.
- ❖ 프로그래머가 프로그램을 더 장황하게 작성할수록, 생산성은 더욱 높아짐
 - 코드의 라인 수에 기반한 생산성 측정은 장황한 코드를 작성하는 프로그래머가 간결한 코드를 작성하는 프로그래머보다 생산성이 높은 결과를 제시함.

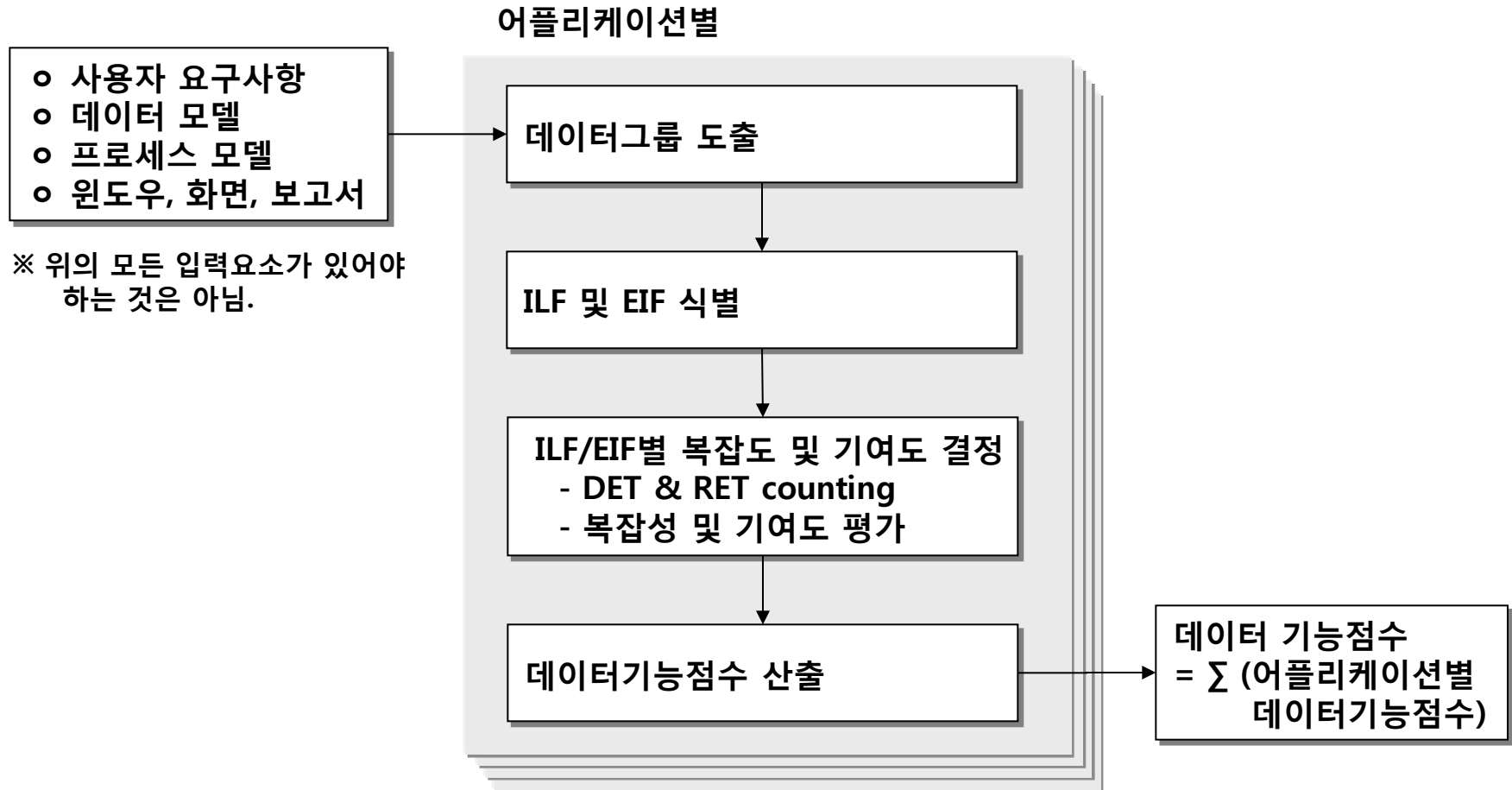
기능 점수



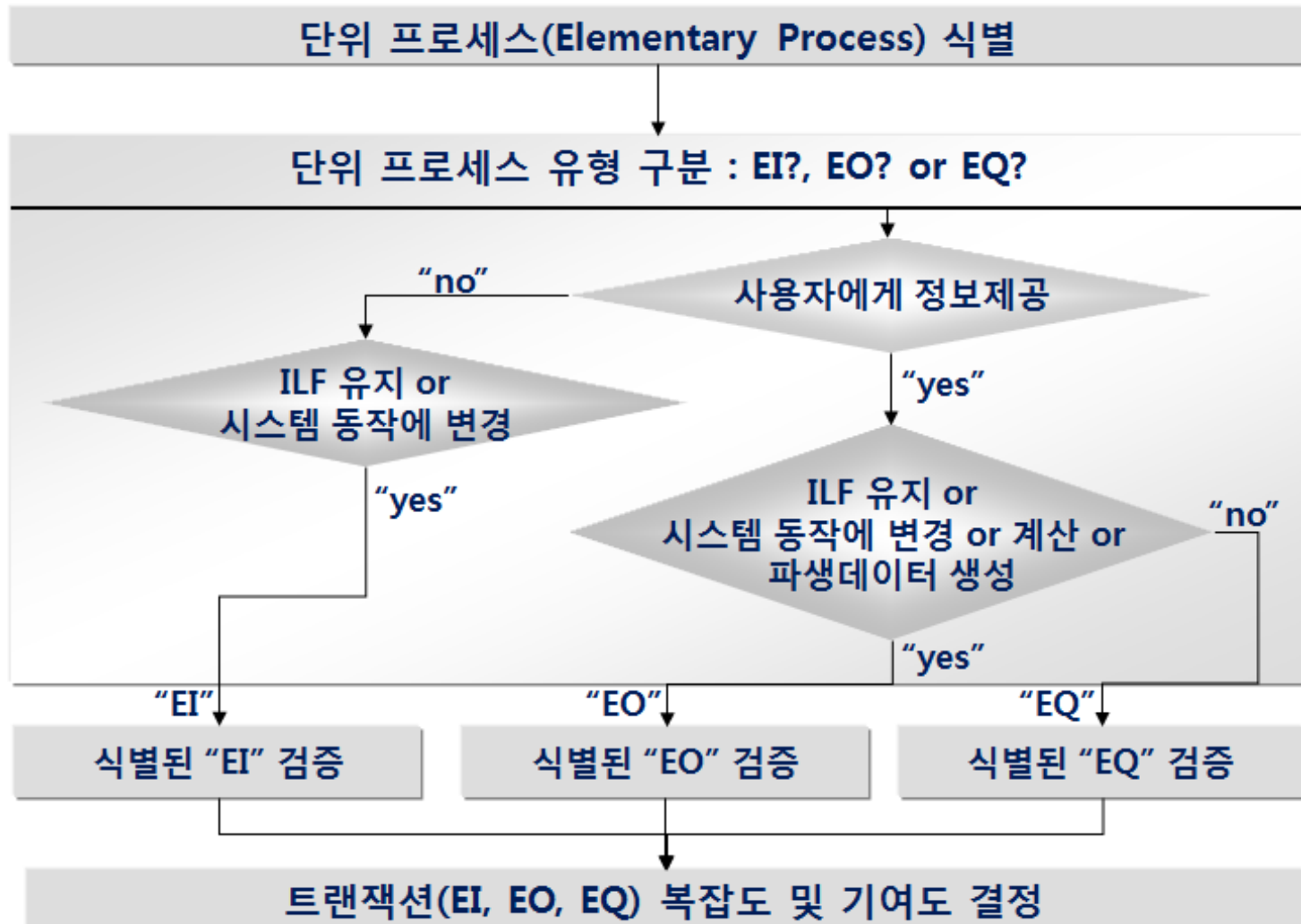
기능 점수 계산 과정



데이터 기능 점수



트랜잭션 기능점수



기능 요소별 복잡도 (1/2)

[표 3-2] ILF, EIF의 복잡도

RET 개수	DET 개수		
	1~19	20~50	51이상
1	낮음	낮음	보통
2~5	낮음	보통	높음
60이상	보통	높음	높음

[표 3-3] EI의 복잡도

FTR 개수	DET 개수		
	1~4	5~15	16이상
0~1	낮음	낮음	보통
2	낮음	보통	높음
30이상	보통	높음	높음

기능 요소별 복잡도(2/2)

[표 3-4] EO, EQ의 복잡도

FTR의 개수	DET 개수		
	1~5	6~19	20이상
0~1	낮음	낮음	보통
2~3	낮음	보통	높음
4이상	보통	높음	높음

[표 3-5] 복잡도별 가중치

기능요소 \ 복잡도	낮음	보통	높음
ILF	7	10	15
EIF	5	7	10
EI	3	4	6
EO	4	5	7
EQ	3	4	6

기능 점수 계산

[표 3-6] 복잡도별 기능 요소의 개수

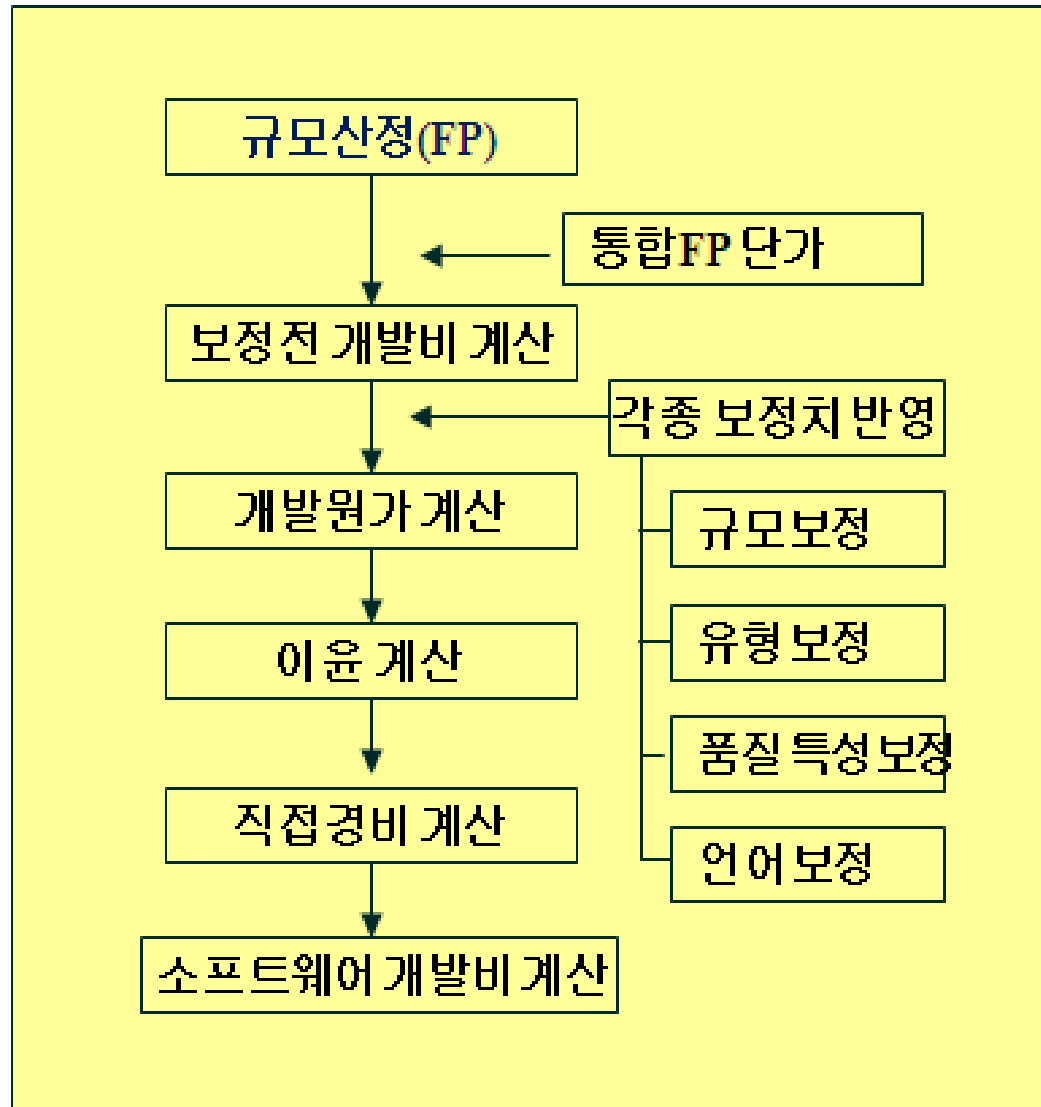
기능요소 \ 복잡도	낮음	보통	높음
ILF	3	3	6
EIF	4	5	5
EI	10	9	15
EO	14	12	14
EQ	2	4	7

[표 3-7] 기능 점수의 계산

기능요소 \ 복잡도	낮음	보통	높음
ILF	$3 \times 7 = 21$	$3 \times 10 = 30$	$6 \times 15 = 90$
EIF	$4 \times 5 = 20$	$5 \times 7 = 35$	$5 \times 10 = 50$
EI	$10 \times 3 = 30$	$9 \times 4 = 36$	$15 \times 6 = 90$
EO	$14 \times 4 = 56$	$12 \times 5 = 60$	$14 \times 7 = 98$
EQ	$2 \times 3 = 6$	$4 \times 4 = 16$	$7 \times 6 = 42$

$$21+30+90+20+35+50+30+36+90+56+60+98+6+16+42=680$$

소프트웨어 사업대가 산정 가이드



다양한 소프트웨어 비용 산정 기법

기법	설명
알고리즘 비용 산정 모델	소프트웨어 크기 척도와 프로젝트 비용과 관련된 과거의 정보를 사용하여 모델이 개발된다. 크기 척도에 대한 추정이 행해지고 모델은 소프트웨어 개발 비용을 추정한다.
전문가의 판단	제안된 소프트웨어 개발 기법과 응용 분야에 대한 전문가의 자문을 구한다. 전문가 각각이 프로젝트 비용을 추정한다. 이들 추정값들은 비교되고 토의된다. 일치된 추정값을 얻을 때까지 반복된다.
유추에 의한 산정	이 기법은 동일한 응용 분야에 대한 유사한 프로젝트의 비용 정보가 있을 때 적용 가능하다. 새로운 프로젝트의 비용은 이들 완성된 프로젝트를 가지고 유추에 의해 추정된다.
Parkinson의 법칙	Parkinson의 법칙은 작업은 가능한 시간을 다 채울 때까지 확장된다는 것을 말한다. 비용은 객관적인 평가에 의한 것이 아니라 이용 가능한 자원에 의해 결정된다.
합의에 의한 가격 결정	소프트웨어 가격은 고객이 그 프로젝트에 지불할 수 있는 만큼의 비용으로 가격이 결정된다. 추정된 노력은 소프트웨어의 기능에 좌우되는 것이 아니라 고객의 예산에 좌우된다.

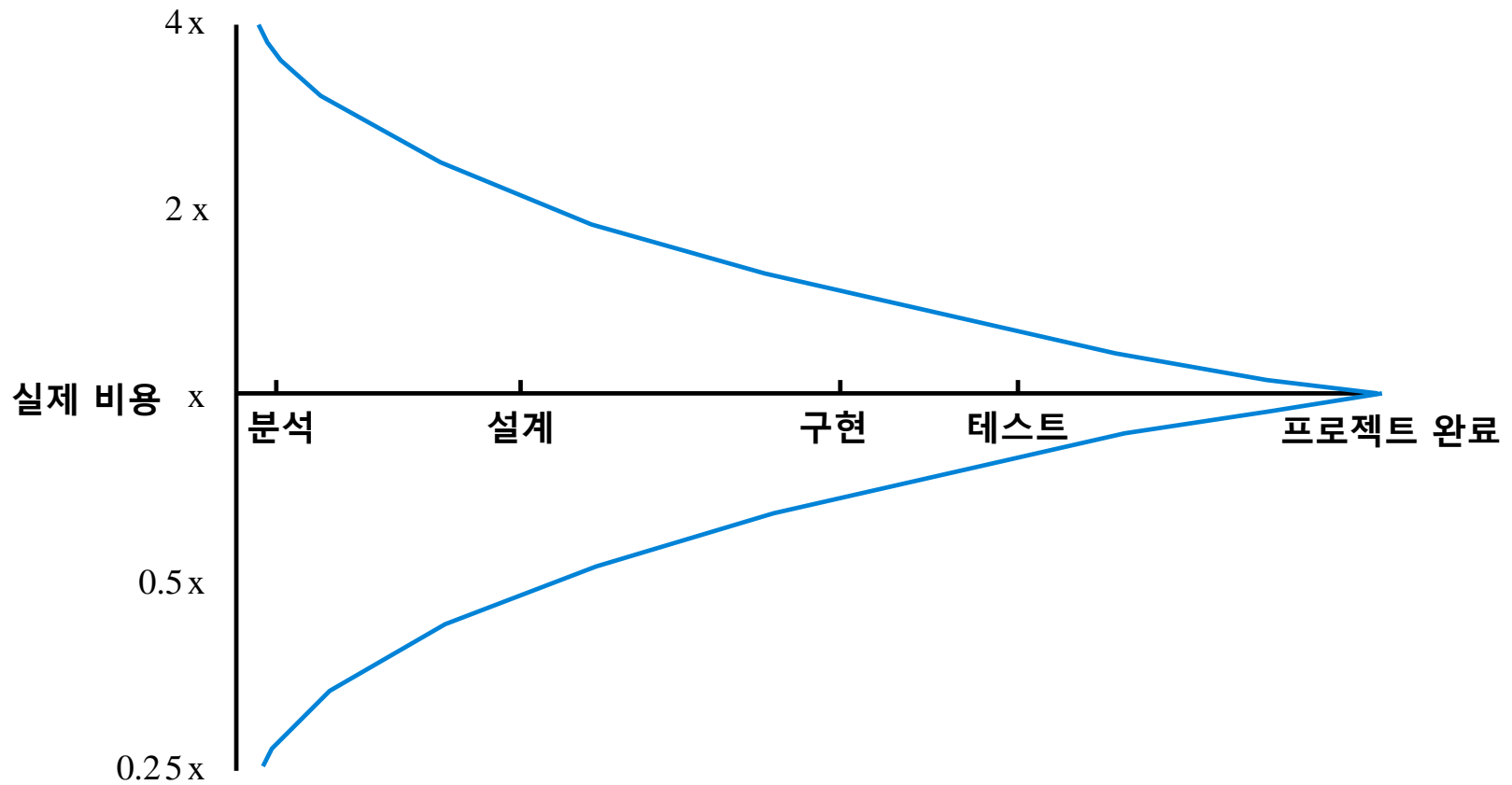
알고리즘 비용 산정 모델

- ❖ 비용은 관리자가 추정하는 제품, 프로젝트, 프로세스 속성의 수학 함수를 이용하여 추정됨:
 - $\text{Effort} = A \times \text{Size}^B \times M$
 - A는 상수, Size는 규모 추정치, B는 규모의 비경제 반영 지수, M은 제품 프로세스 인간 속성을 반영하는 乘數.
- ❖ 비용 산정을 위해 가장 공통적으로 이용되는 속성은 코드 규모.
- ❖ A, B, M은 대부분의 비용 산정 모델이 유사하기는 하나 상이한 값을 가짐.

비용 산정의 정확도

- ❖ 소프트웨어 시스템의 규모는 프로젝트가 종료될 때에만 정확하게 알 수 있음
- ❖ 여러 가지 요인이 최종 규모에 영향을 줌
 - COTS와 컴포넌트의 이용
 - 프로그래밍 언어
 - 시스템 분산
 - 제품, 컴퓨터, 개발 인력, 프로젝트
- ❖ 개발 프로세스가 진행됨에 따라 규모 추정이 더욱 정확하게 됨

추정의 부정확성



COCOMO

Who are COCOMO?

A tribe in Kenya

KBS2 – “an exploration party to challenge the globe”
Sep. 4, 2005



COnstructive COst MOdel

- ❖ Barry Boehm에 의해 1981년에 발표
- ❖ 가장 널리 이용되는 알고리즘 모델(parametric model)
- ❖ 시스템 규모를 주요 effort driver로 포함하는 식을 제공
- ❖ 규모는 thousands of delivered source instructions (KDSI)으로 측정
- ❖ 공수(Person Month) = $a(KDSI)^b$
- ❖ 프로젝트 유형에 따라 적용 식이 다름
 - Organic, Semi-detached, Embedded
- ❖ 정확도 향상과 단계별 추정이 가능한 모델 제공
 - Basic, Intermediate, Detailed



COCOMO 모델

- ❖ 기존 프로젝트의 경험을 토대로 개발된 모델
- ❖ 문서화가 잘 되어있고, 특정 소프트웨어 벤더에 한정되지 않은 공개 모델
- ❖ 19081년에 발표된 초기 버전 (COCOMO-81) 부터 COCOMO 2 까지의 오랜 역사를 가짐
- ❖ COCOMO 2는 소프트웨어 개발, 재사용 등에 상이한 접근법을 적용

COCOMO 81

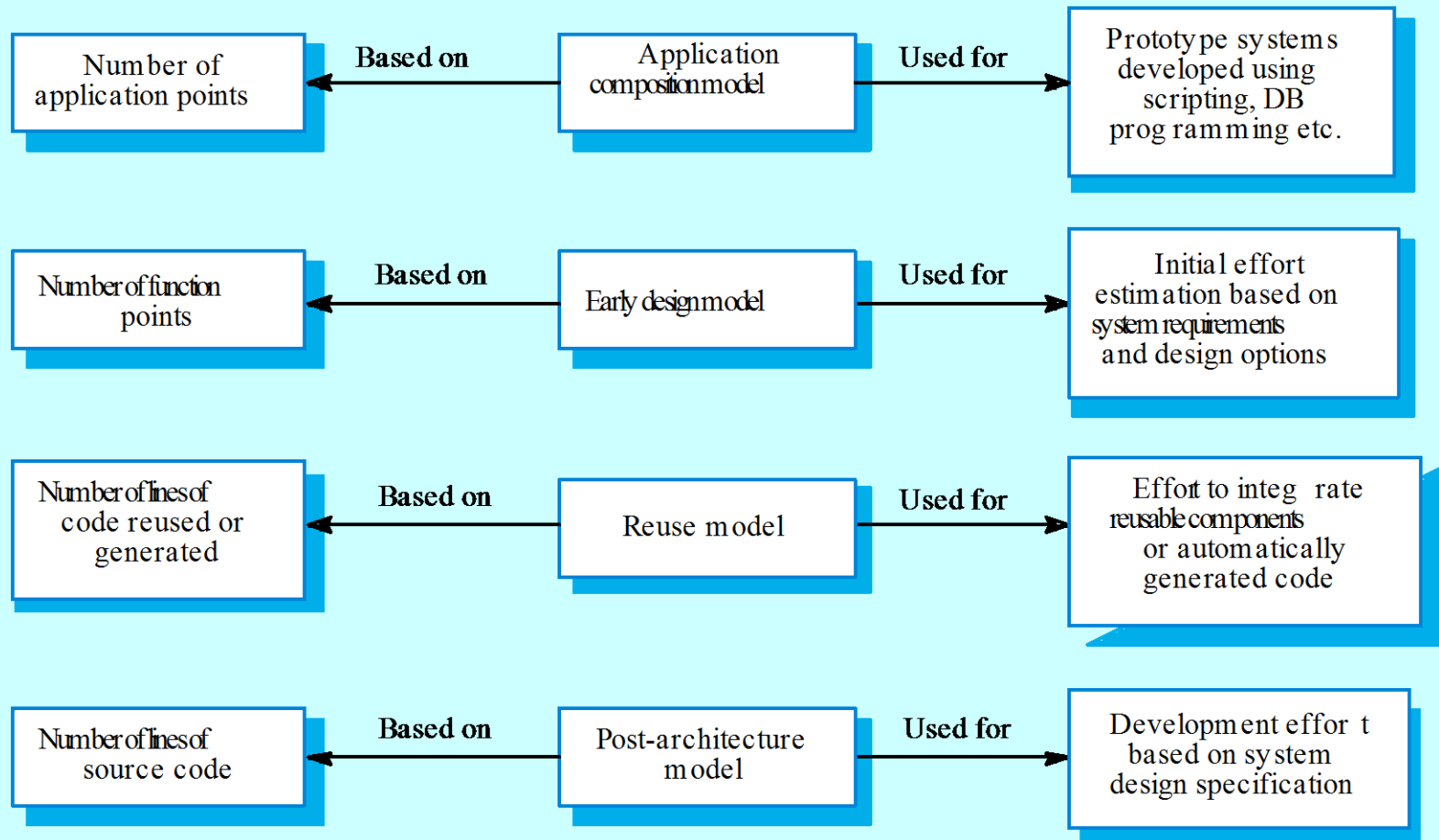
[표 3-8] 기본적인 COCOMO 모델

프로젝트 복잡도	수식	설명
단순(organic)	$PM = 2.4(KDSI)^{1.05} \times M$	소규모 팀에 의해 개발되는 잘 이해된 응용시스템
보통(semidetached)	$PM = 3.0(KDSI)^{1.12} \times M$	팀 구성원들이 개발 시스템에 관련된 제한적인 경험을 가진 더욱 복잡한 시스템
복잡(embedded)	$PM = 3.6(KDSI)^{1.20} \times M$	소프트웨어가 하드웨어, 소프트웨어, 규정, 운영 절차들이 강하게 결합된 복합물의 일부인 복잡한 프로젝트

COCOMO 2

- ❖ COCOMO 81은 폭포수 프로세스가 사용되고, 모든 소프트웨어가 원점에서 개발되는 것을 가정하였음
- ❖ 최초 발표된 후에 소프트웨어 공학의 실무 관행이 많은 변화가 있었기 때문에, COCOMO 2는 소프트웨어 개발을 위한 상이한 접근법들을 수용하기 위해 개발됨

COCOMO 2 서브모델의 이용



초기 설계 모델

- ❖ 요구사항이 도출된 후에 추정이 가능함
- ❖ 알고리즘 모델의 표준 형식에 기반을 둠
 - $PM = A \times Size^B \times M$
 - $M = PERS \times RCPX \times RUSE \times PDIF \times PREX \times FCIL \times SCED$
 - $A = 2.94$, Size는 KLOC, B는 1.1부터 1.24 사이에서 결정(novelty, flexibility, risk resolution process, cohesion, maturity에 따라).

乗數 (Multipliers)

❖ 승수는 개발자, 비기능적 요구사항, 개발 플랫폼의 친숙성 등을 반영

- RCPX - product reliability and complexity
- RUSE - the reuse required
- PDIF - platform difficulty
- PREX - personnel experience
- PERS - personnel capability
- SCED - required schedule
- FCIL - the team support facilities

포스트 아키텍처 모델

- ❖ 초기 설계 모델과 동일한 식을 사용하지만, 승수가 17개
- ❖ 코드 규모의 추정 방식
 - 코드의 라인 수
 - 새로운 코드의 라인 수
 - 재사용 모델을 이용하여 계산된, 새로운 코드에 해당하는 라인 수
 - 요구사항 변경으로 인해 수정해야 하는 코드의 라인 수
 - 기능 점수

지수항 (B)

- ❖ 스케일 인자의 합/100 을 1.01에 더함
- ❖ 예: CMM 레벨 2
 - Precedentness - new project (4)
 - Development flexibility - no client involvement - Very high (1)
 - Architecture/risk resolution - No risk analysis - V. Low (5)
 - Team cohesion - new team - nominal (3)
 - Process maturity - some control - nominal (3)
- ❖ 따라서 B는 1.17

승수 (비용 인자)

속성	유형	설명
RELY	제품	요구되는 시스템 신뢰성
CPLX	제품	시스템 모듈의 복잡성
DOCU	제품	요구되는 문서화 정도
DATA	제품	사용된 데이터베이스 크기
RUSE	제품	요구되는 재사용 컴포넌트의 백분율
TIME	컴퓨터	실행 시간 제약
PVOL	컴퓨터	개발 플랫폼의 가변성
STOR	컴퓨터	메모리 제약
ACAP	인적	프로젝트 분석가 능력
PCON	인적	인적 연속성
PCAP	인적	프로그래머 능력
PEXP	인적	프로젝트 분야에 대한 프로그래머 경험
AEXP	인적	프로젝트 분야에 대한 분석가 경험
LTEX	프로젝트	언어와 도구 경험
TOOL	프로젝트	소프트웨어 도구의 사용
SCED	프로젝트	개발 일정 제약
SITE	프로젝트	다중 사이트 작업 정도 및 사이트 간의 통신 수준

비용 인자의 영향

지수 값	1.17
시스템 크기	128, 000 LOC
비용인자 없는 초기의 COCOMO 추정	730 person-months
신뢰성	매우 높음, 승수 = 1.39
복잡성	매우 높음, 승수 = 1.3
메모리 제한	높음, 승수 = 1.21
도구 사용	낮음, 승수 = 1.12
일정	단축, 승수 = 1.29
조정된 COCOMO 추정	2306 person-months
신뢰성	매우 낮음, 승수 = 0.75
복잡성	매우 낮음, 승수 = 0.75
메모리 제한	없음, 승수 = 1
도구 사용	매우 높음, 승수 = 0.72
일정	보통, 승수 = 1
조정된 COCOMO 추정	295 person-months

