IT@COOKBOOK

쉽게 배우는 소프트웨어 공학

2판

Chapter 03 계획

목차

학습목표

- 체계적인 계획을 세우는 방법을 알아본다.
- 소프트웨어 개발 비용 산정을 알아본다.
- 기능 점수 방법을 자세히 살펴본다.
- 프로젝트 일정 계획과 위험 분석 방법을 알아본다.

01. 계획의 이해

■ 계획

▪ 계획의 중요성

- 소프트웨어 개발의 성패는 비용, 기간, 인력과 같은 자원을 토대로 초기에 얼마나 **계획을 잘 세우느냐에** 달려있음
- 계획을 제대로 세우지 않고 수행하는 소프트웨어 개발은 '일정지연, 비용초과, 품질저하' 등이 발생할 수 있으며, 유지 보수 비용 증가 결과를 불러 옴.
- 계획을 잘 세우려면 '명확한 사용자 요구 사항'과 '충분한 정보', '제약 조건의 충분한 이해'가 필수

02. 문제 정의

■ 문제 정의

- 문제 정의
  - SW 개발의 첫 단계.
  - 무엇을 개발할 것인가에 대한 명확한 정의 필요. => 개발 범위 결정

- 문제 정의 시 필요한 사항
  - 개발 영역의 배경 지식
  - 현재 운영 중인 시스템 사용 경험 //아날로그 시스템 일 수도 있음
  - 실무 담당자와 면담 및 자료 확보를 통한 분석
  - 유사한 프로젝트 개발 경험의 분석가가 참여하는 것도 도움이 됨

03. 타당성 분석

■ 타당성 분석

▪ 경제적 타당성

- 경영자 입장에서 의사결정을 하는 데 매우 중요한 요소
- 시장 분석을 통해 시장성을 확인
- 경제적 타당성 분석으로 투자 효율성과 시장성을 검증한 후 개발 여부를 판단

▪ 기술적 타당성

- 요구하는 기술을 회사가 가지고 있는지 확인
- 사용자가 요구하는 프로젝트가 최신 기술이 필요하다면 기술적 타당성을 먼저 검토
- 부족하다면 필요한 기술을 갖고 있는 소프트웨어 개발자를 채용하거나 외주 개발로 해결

▪ 법적 타당성

- 오픈소스는 소프트웨어 개발에서 분쟁 발생 소지가 높음
- 소프트웨어 개발에서 오픈소스를 사용하는 것은 비용 절감 측면에서 매우 효율적
- 오픈소스는 원시 코드가 개방되어 있다는 것이지 아무렇게나 가져다 사용할 수 있는 것은 아님
- 법적인 문제가 발생하지 않으려면 오픈소스를 사용 할 때 어디까지 무료로 사용할 수 있는지 확인해야 함

04. 개발 비용 산정

■ 소프트웨어 개발 비용 산정의 어려움

- 소프트웨어 가격은 근거를 명확히 제시하기 어려우니 사용자도 받아들이기가 쉽지 않음
- 개발 프로세스가 다양하기 때문에 표준화나 자동화가 어려워 개발 프로세스에 따라 생산성이나 품질이 서로 다를 수 있음

- 조립 컴퓨터 가격, 전자 제품 가격, 건축비 등은 어느 정도 근거가 명확해 소비자도 인정하기가 쉬움

부품	수량	부품 단가	합계	인건비	조립 컴퓨터 생산 원가
CPU	1개	232,000원	979,130원	200,000원	1,179,130원
메인보드	1개	244,000원			
RAM	2개	24,000원			
VGA	1개	277,500원			
SSD	1개	140,000원			
HDD	1개	39,800원			
ODD	1개	21,830원			

그림 3-3 조립 컴퓨터 생산 원가 산정의 예

04. 개발 비용 산정

■ 개발 비용에 영향을 주는 요소

- 프로그래머 자질
  - 초급 프로그래머와 경험이 많은 프로그래머의 생산성에는 큰 차이가 있음
- 소프트웨어 복잡도
  - 브룩스의 법칙: "애플리케이션을 개발하는 것보다 유틸리티를 개발하는 것이 세 배 어렵고, 유틸리티를 개발하는 것보다 시스템 프로그램을 개발하는 것이 세 배 어렵다"
- 소프트웨어 크기
  - 개발하려는 소프트웨어의 규모가 크면 개발 인력과 개발 기간도 늘고 복잡도도 더 커짐
- 가용 시간
  - 관리자의 잘못된 생각 중 하나가 개발 기간을 단축하려면 인력과 자원을 늘리면 된다는 것
  - 보엠: "인력과 자원을 늘려도 정상적인 계획에서 최대 75%가 줄일 수 있는 한계"
- 요구되는 신뢰도 수준
  - 사고나 오류 발생 시 재산에 큰 손실을 끼치거나, 인명 피해가 발생할 수 있는 소프트웨어들은 개발 시 높은 신뢰도를 요구
  - 대학종합정보시스템 : 금융SW/의료장비SW/미사일SW
- 기술 수준
  - 소프트웨어 개발 시 고급 언어를 사용하면 저수준 언어를 사용할 때보다 프로그래머의 생산성이 5~10배 높아짐

05. 비용 산정 기법 1: 하향식 산정 기법

■ 하향식 비용 산정 기법 : 과거 경험 바탕으로 비용 산정

▪ 전문가 판단 기법

- 경험이 많은 여러 전문가가 프로젝트를 수행하는 데 비용이 어느 정도 들어가는지 평가한 금액을 개발 비용으로 산정
- 경험이 많은 전문가가 판단 을 내린 만큼 신뢰성이 있고 편리하다는 장점
- 짧은 시간에 개발비를 산정하거나 입찰에 응해야 하는 경우 많이 사용
- 수학적 계산에 의해 산정하지 않고 경험에만 의존할 경우 부정확할 수 있음

▪ 델파이 기법

- 전문가의 경험을 중요시해 비용을 산정하는 것은 같으나 전문가들의 편견이나 분위기에 영향을 받지 않도록 조정자를 둠
- 여러 전문가가 모여 각자의 의견대로 비용을 산정 후 결과를 공유하고 의견을 조율하여 개발 비용을 산정

① 조정자는 전문가가 모여 비용 산정을 하는 회의에서 간사 역할을 수행
② 전문가는 비용을 산정할 수 있는 자료를 충분히 검토하고, 필요하다면 의견을 나눌 수 있음
③ 전문가 각자가 비용을 산정한다. 이때 계산된 결과를 조정자에게 익명으로 제출
④ 조정자는 각 전문가가 제출한 자료를 요약 정리
⑤ 조정자는 각 전문가가 제출한 자료에서 산정 내용에 차이가 크면 이 문제를 해결하기 위해 회의를 소집
⑥ 전문가는 다시 익명으로 비용 산정 작업을 실시

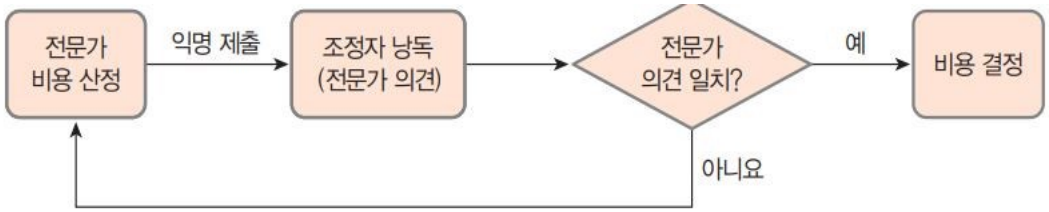


그림 3-4 델파이 기법의 비용 산정 방법

06. 비용 산정 기법 2: 상향식 산정 기법

■ 상향식 비용 산정 기법

▪ 원시 코드 라인 수(line of code : LOC) 기법

- 소프트웨어 각 기능의 개발 비용을 산정할 때 원시 코드의 크기, 즉 라인 수를 중심에 두는 방법
- 먼저 완성될 소프트웨어의 크기(라인 수LOC)를 추정하고 이를 준비된 식에 대입해 개발에 필요한 M/M을 예측

▪ LOC 기법

- 소프트웨어 코드 라인 수를 '비관치/낙관치/중간치'를 이용하여 예측치를 구하고 이를 이용 하여 노력, 개발 비용, 개발 기간, 생산성을 산정하는 방법

- 쇼핑몰 시스템
  - 모듈 1: 상품 관리 모듈
  - 모듈 2: 상품 주문 모듈
  - 모듈 3: 상품 주문 처리 모듈
  - 모듈 4: 회원 관리 모듈

모듈번호	낙관적 LOC (V_{opt})	보통의 LOC (V_m)	비관적 LOC (V_{pess})	추정 LOC (EV)
1	250	400	750	433
2	300	450	820	487
3	350	600	1,100	642
4	170	300	550	320
추정 LOC 합계				1,882

$$EV = (V_{opt} + 4 * V_m + V_{pess}) / 6$$

06. 비용 산정 기법 2: 상향식 산정 기법

■ 상향식 비용 산정 기법

- LOC 관련 식

  - 노력(인/월수$^{M/M}$) = (참여 인원/월) × 개발 기간 = 추정 LOC/1인당 월평균 생산 코드 라인 수
  - 개발 비용 = (M/M) × 단위 비용(1인당 월평균 인건비)
  - 개발 기간 = (M/M)/참여 인원
  - 생산성 = LOC/(M/M)

  - 노력(Effort) 단위 : M/M(Man/Month), P/M(Person/Month).
  - 노력 계산 : 월당 참여인원 * 개발 기간. (프로그램 코드 전체 line) / (1인당 월평균 생산 line)
  - 생산성 : . (프로그램 코드 전체 line) / 노력(M/M, P/M)

- 예

  1. 소프트웨어 개발 기간이 1년 이다. 5명은 12개월 동안, 7명은 5개월 동안 참여한다면 이 SW의 개발 노력(M/M, P/M)은 얼마인가?
     - 5명 * 12개월 + 7명 * 5개월 = 95 (M/M, P/M)

  2. 개발 예측된 코드의 길이가 50,000 line이고, 개발자가 10명 참여한다. 개발자당 월평균 500 line을 코딩 한다면 개발 기간은 얼마인가?
     - 50,000 line / 500 line = 100(M/M)
     - 개발 기간 = 100(M/M) / 10 명

07. 비용 산정 기법 3: 수학적 산정 기법

■ 수학적 상향식 산정 기법

- COCOMO 기법과 Function Point 기법이 있음.

- COCOMO
  - SW 의 규모(LOC)를 예측한 후 SW 종류에 따라 비용 산정 공식에 대입

- Function Point
  - 기능 정수를 구한 후 이를 이용하여 비용 산정

07. 비용 산정 기법 3: 수학적 산정 기법

■ 수학적 상향식 산정 기법

- COCOMO
  - 완성될 SW 의 규모(LOC)를 예측한 후 SW 종류에 따라 비용 산정 공식에 대입
    - 비용 산정은 소스코드 라인 수 중심
    - 비용 산정 공식은 63개 실제 SW 개발 프로젝트를 분석하여 구성함.
    - 1995년에 COCOMO II 로 확장됨.

07. 비용 산정 기법 3: 수학적 산정 기법

■ 수학적 상향식 산정 기법

- COCOMO(COnstructive COst Model) 방법의 고려 사항
  - 3가지 프로그램 유형에 따른 가중치와
  - 특성에 따라 15가지로 분류한 후 인건비를 더 보정

- 유형에 따른 가중치 반영
  - 단순형 프로젝트
    - 복잡도와 난이도가 비교적 높지 않은 업무용 소프트웨어가 이에 해당
    - 중소 규모 정도이고, 크기는 50KDSI(Kilo Delivery Source Instruction) 이하
  - 중간형 프로젝트
    - 일반 업무용 소프트웨어보다 복잡하고 규모가 더 큰 운영체제, 데이터베이스 관리 프로그램 등이 이에 해당
    - 규모나 복잡도가 중간 정도이고, 크기는 300KDSI 이하
  - 내장형(임베디드형) 프로젝트
    - 자동화 기기나 전자 제품과 같은 하드웨어와 밀접하게 관련 있는 내장형 소프트웨어가 이에 해당
    - 크기는 300KDSI 이상

07. 비용 산정 기법 3: 수학적 산정 기법

■ COCOMO 방법

▪ 개발 인건비 초기 예측

- COCOMO 방법에서 제시하는 공식을 사용하면 개발 인건비$^{M/M}$의 초기 예측 값을 계산
- 규모가 같은 소프트웨어일 경우 기본형보다는 중간형에서 , 중간형보다는 내장형에서 M/M이 더 많이 필요

표 3-1 투입 인력 산출 공식과 프로젝트 유형에 따른 상수 a, b값

$PM=a\times(KDSI)^b$	
상수	유형별 값
a	단순형(2.4), 중간형(3.0), 내장형(3.6)
b	단순형(1.05), 중간형(1.12), 내장형(1.20)

표 3-2 프로젝트 유형에 따른 투입 인력 산출 공식

프로젝트 유형	공식
단순형	$PM=2.4\times(KDSI)^{1.05}$
중간형	$PM=3.0\times(KDSI)^{1.12}$
내장형	$PM=3.6\times(KDSI)^{1.20}$

- $PM^{Person/Month}$: 소프트웨어 개발에 필요한 인력(인원/월)
- $KDSI^{Kilo\ Delivered\ Source\ Instruction}$: 소프트웨어의 최종 원시 코드 라인 수

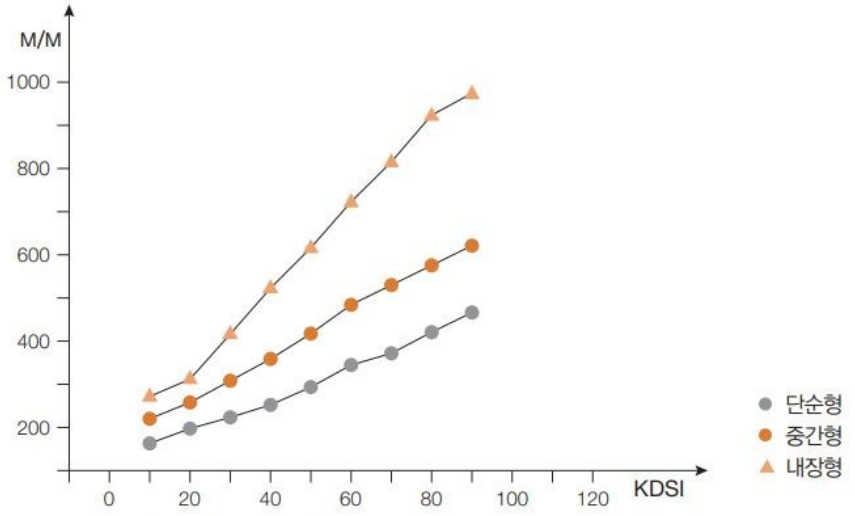


그림 3-5 COCOMO 방법에 의한 비용 예측

07. 비용 산정 기법 3: 수학적 산정 기법

■ COCOMO 방법

▪ 보정 계수 반영하기

- 노력 조정 수치(EAF Effort Adjustment Factor): 보정에 사용하는 값, 필요한 각 항목별 '승수 값'을 모두 곱한 값
- 15개의 세부 항목에 따른 보정 값을 모두 정해 놓음

표 3-3 COCOMO 방법에서 사용되는 노력 승수 값

특성	비용 승수 요소	승수 값					
		매우 낮음	낮음	정상	높음	매우 높음	극히 높음
제품 특성	요구되는 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 특성	실행 시간 제약			1.00	1.11	1.30	1.66
	주기억 장치의 제약			1.00	1.06	1.21	1.56
	HW/SW의 안정성		0.87	1.00	1.15	1.30	
	처리 시간		0.87	1.00	1.07	1.15	
개발자 특성	분석가의 능력	1.46	1.19	1.00	0.86	0.71	
	응용 분야 경험	1.29	1.13	1.00	0.91	0.82	
	컴퓨터와 친숙성	1.21	1.10	1.00	0.90	–	
	프로그래머 능력	1.42	1.17	1.00	0.86	0.70	
	프로그램 언어의 경험	1.14	1.07	1.00	0.95		
프로젝트 특성	소프트웨어 공학 기술 사용	1.24	1.10	1.00	0.91	0.82	
	소프트웨어 도구의 사용	1.24	1.10	1.00	0.91	0.83	
	요구되는 개발 일정	1.23	1.08	1.00	1.04	1.10	

표 3-4 노력 조정 수치가 반영된 유형별 노력 P/M

프로젝트 유형	공식
단순형	$PM = 2.4 \times (KDSI)^{1.05} \times EAF$
중간형	$PM = 3.0 \times (KDSI)^{1.12} \times EAF$
내장형	$PM = 3.6 \times (KDSI)^{1.20} \times EAF$

예1 : 요구되는 신뢰도가 높고(표 3.3 – 1.15), 매우 복잡하며(1.30), 소프트웨어 공학 기술을 거의 사용하지 않고(1.24), 요구되는 개발 일정이 매우 촉박하고(1.10), 다른 요소들은 보통(1.00)일 경우 노력 조정 수치를 계산하라.

EAF : 1.15 * 1.30 * 1.24 * 1.10 = 2.04

예2 : KDSI가 60이고 프로그램 유형은 중간형이며, EAF가 2.04일 때 노력을 구하라.

Effort : 3.0 * (60)^1.12 * 2.04 = 600.179 (P/M)

07. 비용 산정 기법 3: 수학적 산정 기법

■ COCOMO 방법

▪ 총 개발 기간 산정하기

// <주의> LOC 방법과는 다름

- 보엠이 제시하는 총 개발 기간을 계산할 때는 개발할 소프트웨어 유형에 상관없이 모두 동일한 상수(2.5) 값을 곱함
- 개발 기간은 소프트웨어 유형과 상관없음

표 3-5 총 개발 기간 산정 공식

프로젝트 유형	공식
단순형	$TDEV=2.5\times(PM)^{0.38}$
중간형	$TDEV=2.5\times(PM)^{0.35}$
내장형	$TDEV=2.5\times(PM)^{0.32}$

07. 비용 산정 기법 3: 수학적 산정 기법

■ COCOMO II 방법

: 초기에 정확한 LOC 예측의 어려움을 반영한 모델로 단계별 값을 예측 후 인건비 산정에 반영

▪ 애플리케이션 합성 모델

- 1단계에서는 입출력 화면 중심의 사용자 인터페이스 개수 등을 계산해 다음의 객체 점수를 산출
- 개발 범위에 속한 객체(입출력 화면 등)를 찾음
- 객체가 제공하는 기능의 복잡도를 3가지(단순형, 보통형, 복잡형)로 분류
- 객체의 개수에 가중치(단순형, 보통형, 복잡형)를 부여해 결과값을 산출

• 초기 설계 모델

- 2단계는 초기 설계 단계에서 예측값을 구함
- 초기 설계 단계쯤 되면 1단계보다는 시스템의 구조와 기능을 좀 더 상세히 알 수 있어 1단계보다 더욱 정확한 예측이 가능

• 구조 설계 이후 모델

- 3단계에서는 이미 기능 점수가 나왔기 때문에 노력을 추정하는게 어렵지 않음
- 기능 점수를 바탕으로 한 LOC를 추정해 소프트웨어 규모를 산정

• COCOMO II에 적용되는 기본 공식

$$E = b \times S^c \times m(X)$$

$b \times S^c$: 기초 소요 노력 예측값

$m(X)$: 비용 승수의 벡터

07. 비용 산정 기법 3: 수학적 산정 기법

■ 기능 점수 산정 방법

▪ 기능 점수 산정 방법 탄생 배경

- 개발 전에 라인수를 알기도 어렵고, 사용 언어, 개발자, 알고리즘에 따라 라인 수가 달라짐.
- 특히 객체지향 SW는 라인수로 규모 측정이 어려움.

- 규모가 큰 SW는 입출력, 조회, 검색 기능이 많음
- 기능 간 interface나 DB table 수도 많음.
- 프로그램 라인 수 보다 기능이 더 중요한

• SW 기능

- 구현하고자 있는 SW에 있는 입출력, DB table, interface, 조회, 검색 등의 요소

07. 비용 산정 기법 3: 수학적 산정 기법

■ 기능 점수 산정 방법

▪ 기능 점수 산정 방법

• 기능 점수 산정 방법의 개요

- 사용자 관점에서 소프트웨어의 기능을 정량화해 소프트웨어 개발 비용 산정
- 소프트웨어 기능의 복잡한 정도를 상대적인 점수로 표현
- 사무용 또는 관리용 소프트웨어에서 매우 유용한 방법 // Embedded SW에는 맞지 않음.

• 기능 점수 산정 방법의 용도

- SW 개발 시 비용 산정
- SW 유지보수 시 비용 산정
- SW 개발 시 필요한 자원 산정

• 기능 점수 산정 방법의 특징

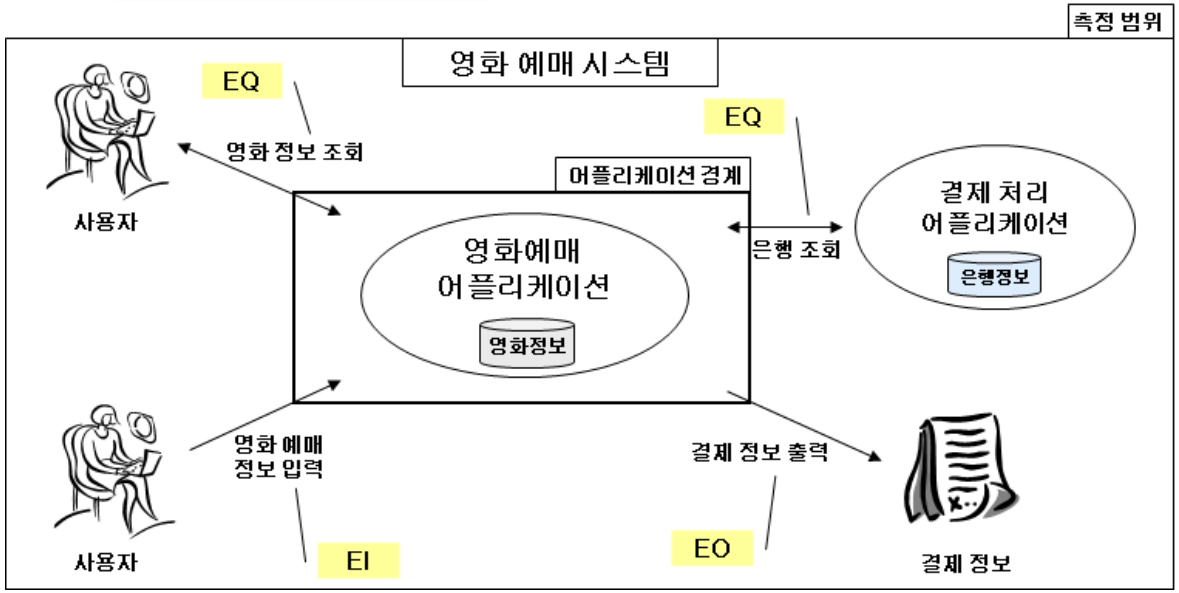
- 기능적 요구사항 중심의 측정 방법
- 소프트웨어의 요구사항 복잡도 측정 방법
- 구현 관점(물리적 파일, 화면, 프로그램 수)이 아닌 사용자 관점의 요구 기능을 정량적으로 산정
- 측정의 일관성을 유지하기 위해 개발 기술, 개발 방법, 품질 수준 등은 고려하지 않음
- 소프트웨어 개발에 사용하는 언어와 무관
- 소프트웨어 개발 생명주기의 전체 단계에서 사용 가능

07. 비용 산정 기법 3: 수학적 산정 기법

■ 기능 점수 산정 방법

- 기능 점수 산정 방법의 구분
  - 기능 점수의 기준이 되는 소프트웨어 기능은 크게 데이터 기능과 트랜잭션으로 구분

그림 3-6 소프트웨어 기능 분류



07. 비용 산정 기법 3: 수학적 산정 기법

■ 기능 점수 산정 방법

▪ 기능 점수 산정 방법의 구분

- SW사업 대가산정 가이드에서 제시하는 기능 점수 산정 방법

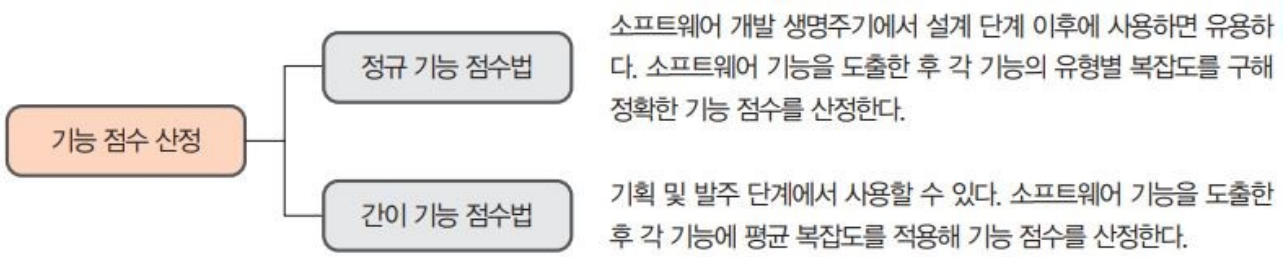


그림 3-7 기능 점수 산정 방법

- 정규 기능 산정 방법 :
  - 파일 내 레코드 단위까지 알아야 함. => 설계 단계 이후에 이용.
  - 각 기능 도출 후 기능의 유형별 복잡도를 구해 기능 점수 산정.
- 간이 기능 점수 방법
  - 파일 레코드 단계까지 알 필요가 없다. => 기획 및 발주 단계에서 사용하면 유용
  - SW 기능을 도출 후 각 기능에 평균 복잡도를 적용해 기능 점수 산정. // 정규 기능 산정 시는 유형별 복잡도 이용

07. 비용 산정 기법 3: 수학적 산정 기법

■ 기능 점수 산정 방법

▪ 기능 점수 산정 방법

• 기능 점수 산정 방법의 장점

- 사용자의 요구사항만으로 기능을 추출해 측정 // 구현 기술, 언어, 도구, 능력 등 실제 구현 방법과 관계 없음.
- 객관적인 요구사항만으로 측정 // 개발 방법, 개발 팀에 독립적이므로 규모 측정에 일관성 유지
- 계획, 분석, 설계 등 모든 개발 단계에서 사용 // 단계가 진행됨에 따라 더 정확한 기능 점수 측정 가능

• 기능 점수 산정 방법의 단점

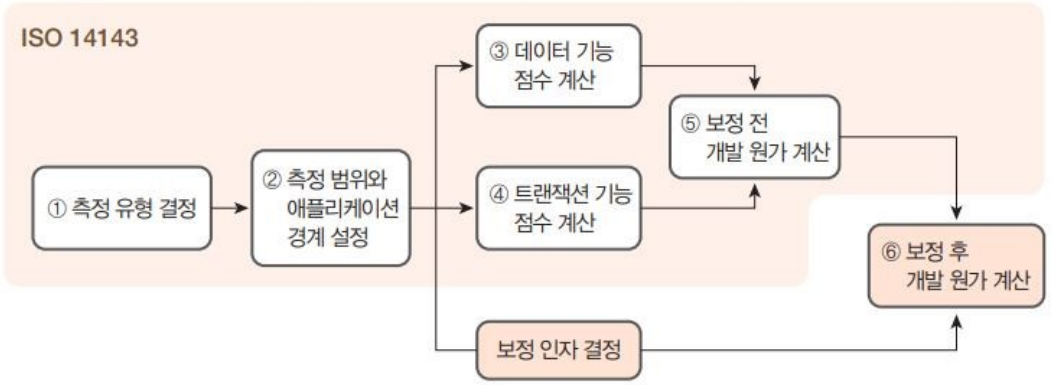
- 높은 분석 능력 필요 // 요구사항에서 기능 도출을 위한 상당한 분석능력이 요구됨.
- 기능 점수 전문가 필요
- 내부 로직 위주의 소프트웨어에는 다소 부적합
- 개발 공수보다 개발 규모 측정에 적합

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 간이 기능 점수법

- 프로젝트 초기 단계에서 각 기능의 요소를 모르는 경우에 평균 복잡도 가중치를 사용해 소프트웨어 기능의 크기를 측정



① 측정 유형(개발, 개선, 애플리케이션) 결정

② 개발하려는 소프트웨어의 측정 범위와 애플리케이션 경계 설정

③ 데이터 기능 점수 계산: 데이터 기능(내부 논리 파일, 외부 연계 파일) 도출 → 도출된 데이터 기능(내부 논리/외부 연계 파일)의 개수×평균 복잡도 가중치

④ 트랜잭션 기능 점수 계산: 트랜잭션 기능(외부 입출력, 외부 조회) 도출 → 도출된 트랜잭션 기능(외부 입출력, 외부 조회)의 개수×평균 복잡도 가중치

⑤ 보정 전 개발 원가 계산: 총 기능 점수×기능 점수당 단가(총 기능 점수 = 데이터 기능 점수+트랜잭션 기능 점수)

⑥ 보정 후 개발 원가 계산: 보정 전 개발 원가×4가지 보정 계수(규모, 애플리케이션 유형, 언어, 품질 및 특성)

그림 3-8 간이 기능 점수 산정 절차

출처 : CPM 4.1, IFPUG, 1999

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 간이 기능 점수법

• 평균 복잡도 가중치

- 과거에 수행한 소프트웨어의 기능 점수 산정 결과를 분석해 5가지 유형에 적용된 복잡도를 계산한 가중치의 평균값
- 사업 초기에는 기능 점수를 산정할 수 있을 만큼 자료가 충분하지 않으므로 평균 복잡도 가중치에 의존해 기능 점수를 산정
- 정상적인 기능 점수 산정 결과를 검증할 때도 평균 복잡도 가중치를 적용해 기능 점수를 산정

표 3-6 평균 복잡도 가중치

유형	내부 논리 파일	외부 연계 파일	외부 입력	외부 출력	외부 조회
가중치	7.5	5.4	4.0	5.2	3.9

출처 : SW사업 대가산정 가이드(2020년 개정판), 한국소프트웨어산업협회

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

- 측정 유형 결정
  - 개발 프로젝트 기능 점수(개발 규모 측정)
    - 프로젝트가 완료되어 최초 설치했을 때의 기능 크기를 산정하는 것
    - 프로젝트에서 사용자를 위해 제공된 모든 기능을 측정
  - 개선 프로젝트 기능 점수(변경 규모 측정)
    - 사용 중인 소프트웨어에 변경이 발생했을 때 변경된 부분의 기능을 측정
    - 완료된 프로젝트에서 추가, 수정, 삭제된 부분만 그 크기를 측정
  - 애플리케이션 기능 점수(응용 규모 측정)
    - 애플리케이션 기능 점수는 현재 운용 중인 애플리케이션의 기능을 측정
    - 개발 프로젝트 기능 점수에 개선 프로젝트 기능 점수까지 포함

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 측정 범위와 애플리케이션 경계 설정

- 측정 범위에 포함될 요소(서브시스템)의 식별
  - 측정 범위: 기능 점수를 측정하고자 하는 대상

- 애플리케이션 경계
  - 애플리케이션 경계: 기능 점수를 계산하는 대상을 제외한 다른 애플리케이션이나 외부 사용자를 구분한 경계
  - 경계를 지을 때 주의할 점은 사용자 관점에서 구분해야함(개발자 관점에서 구분해서는 안됨)
  - 애플리케이션의 경계도 적당해야 함

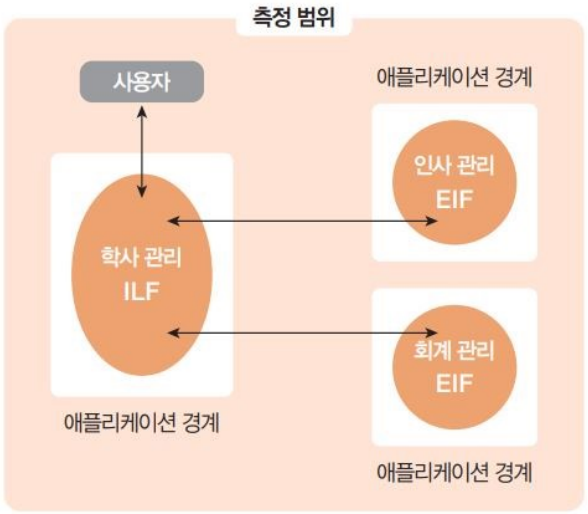


그림 3-9 대학 종합정보시스템의 애플리케이션 경계

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 데이터 기능 점수 계산

- 내부 논리 파일(ILF)의 개수와 외부 연계 파일(EIF)의 개수를 계산해 각각의 평균 복잡도에 따라 데이터 기능 점수가 결정

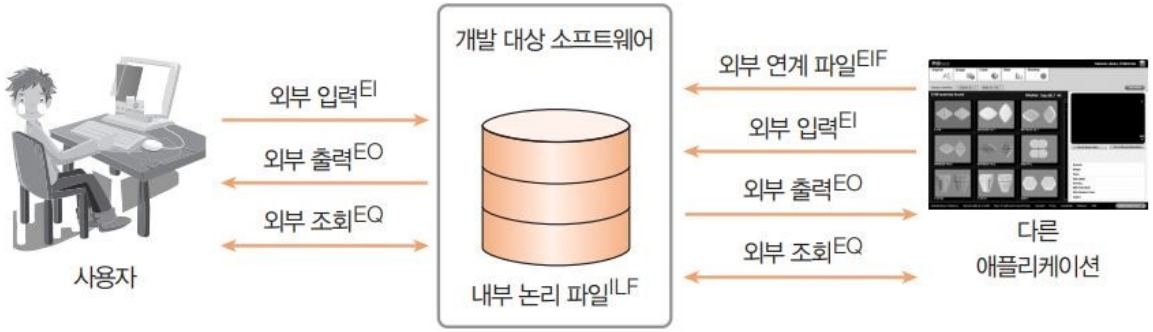


그림 3-10 사용자 관점에서 본 기능 점수의 5가지 유형

- 내부 논리 파일
  - 사용자가 등록/수정/삭제/조회를 하기 위한 대상으로, 데이터베이스에 존재하는 데이터 모임(데이터베이스의 정보)
  - 데이터베이스의 정보들은 기능 점수 측정 대상으로 애플리케이션 내부에서 파일로 유지

- 외부 연계 파일
  - 측정 대상 애플리케이션에서는 참조만 하고 다른 애플리케이션에서 유지되는 파일

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 데이터 기능 점수 계산

- 이번 프로젝트에서 생성해 관리하는 데이터는 내부 논리 파일(ILF)
- 이번 프로젝트에서 만들지 않고 다른 프로젝트에서 생성했으나 이번 프로젝트에서 참조하는 데이터는 외부 연계 파일(EIF)

표 3-7 데이터 기능의 예

기능	기능 유형	기능명	기능 유형
학생 관리	학생정보 관리	학생 기본 정보	내부 논리 파일ILF
		학생 등록금 정보	외부 연계 파일EIF

- 데이터 기능 점수는 내부 논리 파일의 개수와 외부 연계 파일의 개수에 각각의 평균 복잡도(가중치)를 곱해 계산

데이터 기능 점수 = {(ILF 개수×7.5) + (EIF 개수×5.4)}

7.5는 내부 논리 파일의 평균 복잡도, 5.4는 외부 연계 파일의 평균 복잡도

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 트랜잭션 기능 점수 계산

- 기능 점수에서 트랜잭션 기능의 복잡도는 입력, 출력, 조회의 개수로 결정

> 트랜잭션 기능 점수 = {(EI 개수 × 4.0) + (EO 개수 × 5.2) + (EQ 개수 × 3.9)}
>
> 4.0은 외부 입력의 평균 복잡도, 5.2는 외부 출력의 평균 복잡도, 3.9는 외부 조회의 평균 복잡도

- 외부 입력(EI)
  - 데이터베이스에 데이터를 등록하거나 수정·삭제하는 것(학생정보 등록, 수정, 삭제)
- 외부 출력(EO)
  - 계산하는 로직을 거쳐 데이터나 제어 정보를 사용자에게 보여주는 기능(학생 학점 조회)
- 외부 조회(EQ)
  - 로직이 필요 없고 데이터베이스에 존재하는 데이터를 찾아 그대로 표시만 해주는 기능(학생 주소 검색, 학생정보 조회)

표 3-8 트랜잭션 기능의 예

기능명	기능 유형	기능명	기능 유형
학생 관리	학생정보 관리	학생정보 등록	외부 입력EI
		학생정보 수정	외부 입력EI
		학생정보 조회	외부 조회EQ
		학생정보 삭제	외부 입력EI
		학생 학점 조회	외부 출력EO
	비밀번호 관리	비밀번호 조회	외부 조회EQ
		비밀번호 변경	외부 입력EI

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 미조정 기능 점수 계산(UFP)

- 앞에서 구한 데이터 기능 점수와 트랜잭션 기능 점수를 합한 값
- 단순히 기능적인 요구사항에 대해서만 계산했지, 여러 가지 특성에 대한 고려를 하지 않음

표 3-9 미조정 기능 점수의 예

기능		기능 수	평균 복잡도	기능 점수	
데이터 기능 점수	ILF	1	7.5	7.5	12.9
	EIF	1	5.4	5.4	
트랜잭션 기능 점수	EI	4	4.0	16	29
	EO	1	5.2	5.2	
	EQ	2	3.9	7.8	
계				41.9	

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 보정 전 개발 원가 계산

- 미조정 기능 점수에 기능 점수당 단가를 곱해 계산

> 보정 전 개발 원가=미조정 기능 점수×기능 점수당 단가 (기능 점수당 단가: 553,114원)

- 분석/설계와 구현/테스트를 구분해 별도의 사업으로 진행 시 복잡도 가중치와 가중치에 따른 단계별 단가를 적용

표 3-10 소프트웨어 개발 단계별 기능 점수 가중치

단계	분석	설계	구현	테스트	합계
평균 복잡도 가중치	0.19	0.24	0.32	0.25	1.00
단가	105,092원	132,747원	176,996원	138,279원	553,114원

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 보정 계수

- 보정 계수의 필요성
  - 규모와 유형도 다양하고, 연계 복잡성 수준, 성능요구 수준, 운영 환경 호환성, 보안성 수준이 프로젝트의 특성에 따라 다름
  - 실제 개발 비용에 영향을 미치므로 이들에 대 해 보정 계수를 정하고 이를 반영해 보정한 후 개발 원가를 구해야 함

- 규모 보정 계수
  - 기능 점수가 500FP 미만이면 규모 보정 계수 1.28을, 3,000FP 초과이면 1.153을 적용

$$\text{규모 보정 계수} = 0.4057 \times (\log_e(\text{기능 점수}) - 7.1978)^2 + 0.8878$$

(단, 500FP 미만 시 1.2800, 3000FP 초과 시 1.1530을 적용한다.)

- 애플리케이션 복잡도 보정 계수
  - 사용자가 요구하는 소프트웨어가 복잡하다면 어려운 만큼의 보정 계수를 적용
  - 애플리케이션의 복잡도는 연계 복잡성 수준, 성능요구 수준, 운영 환경 호환성, 보안성 수준으로 나뉨

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 보정 계수

• 애플리케이션 복잡도 보정 계수

- 연계 복잡성 수준
  - 연계 기관 수에 따른 보정 계수를 적용

표 3-11 연계 복잡성 수준에 따른 보정 계수

난이도	보정 계수
1. 타 기관과 연계 없음	0.88
2. 1~2개의 타 기관과 연계	0.94
3. 3~5개의 타 기관과 연계	1.00
4. 6~10개의 타 기관과 연계	1.06
5. 10개를 초과하는 타 기관과 연계	1.12

- 성능요구 수준
  - 개발하려는 소프트웨어에 대해 사용자가 빠른 응답시간 또는 높은 처리율을 요구한다면 소프트웨어의 복잡도는 높아짐

표 3-12 성능요구 수준에 따른 보정 계수

난이도	보정 계수
1. 응답 성능에 대한 특별한 요구사항이 없다.	0.91
2. 응답 성능에 대한 요구사항이 있으나 특별한 조치는 필요하지 않다.	0.95
3. 응답시간이나 처리율이 피크peak 타임에 중요하며 처리 시한이 명시되어 있다.	1.00
4. 응답시간이나 처리율이 모든 업무 시간에 중요하며 처리 시한이 명시되어 있다.	1.05
5. 응답성능 요구사항이 엄격해 설계 단계부터 성능 분석이 요구되거나 설계 및 구현 단계에서 성능 분석 도구가 사용된다.	1.09

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 보정 계수

• 애플리케이션 복잡도 보정 계수

- 운영 환경 호환성
  - 개발이 완료되어 설치할 때의 운영 환경(HW/SW)이 단순하지 않고 다음 표처럼 다양하다면 이를 보정 계수로 반영

표 3-13 운영 환경 호환성에 따른 보정 계수

난이도	보정 계수
1. 운영 환경 호환성에 대한 요구사항이 없다.	0.94
2. 운영 환경 호환성에 대한 요구사항이 있으며 동일 하드웨어 및 소프트웨어 환경에서 운영되도록 설계된다.	1.00
3. 유사한 운영 환경에 대한 요구사항이 있으며 유사 하드웨어 및 소프트웨어 환경에서 운영되도록 설계된다.	1.06
4. 상이한 운영 환경에 대한 요구사항이 있으며 이질적인 하드웨어 및 소프트웨어 환경에서 운영되도록 설계된다.	1.13
5. 항목 4에 더해 일반적 산출물 외에 여러 장소에서 원활한 운영을 보장하기 위한 운영 절차의 문서화와 사전 모의훈련이 요구된다.	1.19

- 보안성 수준
  - 보안에 대한 요구가 많을수록 복잡도는 높아지고 이를 다음 표처럼 보정 계수로 반영

표 3-14 보안성 수준에 따른 보정 계수

난이도	보정 계수
1. 암호화 점검, 웹 취약점 점검, 시큐어 코딩, 개인정보 보호 등 1가지 보안 요구사항이 포함되어 있다.	0.97
2. 2가지 보안 요구사항이 포함되어 있다.	1.00
3. 3가지 보안 요구사항이 포함되어 있다.	1.03
4. 4가지 보안 요구사항이 포함되어 있다.	1.06
5. 5가지 이상의 보안 요구사항이 포함되어 있다.	1.08

07. 비용 산정 기법 3: 수학적 산정 기법

■ 간이 기능 점수법을 이용한 기능 점수 산정 방법

▪ 보정 후 개발 원가 계산

보정 후 개발 원가=보정 전 개발 원가×(규모 보정 계수×연계 복잡성 수준 보정 계수×성능요구 수준 보정 계수×운영 환경 호환성 보정 계수×보안성 수준 보정 계수)

- 5가지 보정 계수의 값을 구했다면 이 값을 적용해 보정한 후 개발 원가를 계산할 수 있음

표 3-15 기능 점수 계산을 위한 애플리케이션

애플리케이션	기능	세부 기능
학사관리	학적 관리	학적 자료 등록/수정/삭제/조회
	교과 과정 관리	교과 과정 등록/수정/삭제/조회, 유사/동일 과목 조회
	수업 관리	개설 강좌 등록/수정/삭제/조회
	수강 관리	수강 과목 등록/수정/조회/삭제, 시간표 등록/수정/삭제/조회, 이수 구분 변경
	성적 관리	성적 등록/수정/조회
사용자 관리	교수 정보 관리	교수 정보 등록/수정/삭제/조회
	학생 정보 관리	학생 정보 등록/수정/삭제/조회

표 3-16 학사관리 애플리케이션에 대한 개발비 계산

단계	기능			개수	평균 가중치	기능 점수
❶	데이터 기능	ILF	학적 테이블, 교과 과정 테이블, 개설 강좌 테이블, 수강 과목 테이블, 성적 테이블	5	7.5	37.5
		EIF	교수 정보 테이블, 학생 정보 테이블	2	5.4	10.8
	트랜잭션 기능	EI	학적 자료 등록/수정/삭제, 교과 과정 등록/수정/삭제, 개설 강좌 등록/수정/삭제, 수강 과목 등록/수정/삭제, 시간표 등록/수정/삭제, 이수 구분 변경, 성적 등록/수정	18	4.0	72.0
		EO	유사/동일 과목 조회, 성적 조회	2	5.2	10.4
		EQ	학적 자료 조회, 교과 과정 조회, 개설 강좌 조회, 수강 과목 조회, 시간표 조회, 교수 정보 조회, 학생 정보 조회	7	3.9	27.3
	기능 점수의 합		158			
❷	보정 전 개발 원가		158×553,114원=87,392,012원			
❸	보정 계수		규모(1.28)×연계 복잡성 수준(0.88)×성능요구 수준(0.91)×운영 환경 호환성(1.0)			
	보정 후 개발 원가		87,392,012원×1.28×0.88×0.91×1.0=89,578,909.71원			

08. 일정 계획

▪ 일정 계획의 이해

• 일정 계획

- 소프트웨어를 개발하기 위해 어떤 작업이 필요한지 찾은 후, 이를 진행할 순서를 결정하거나 주어진 개발 기간에 소작업의 개발 기간 및 그들 간의 순서, 필요한 자원 등과 같은 일정을 계획하는 것

표 3-17 일정 계획

<table>
<tr><td>총 개발 기간</td><td>6개월(○○○○. 1. 1 ~ ○○○○. 6. 31)</td></tr>
<tr><td>소작업</td><td>교과 과정 관리, 수업 관리, 수강 관리, 성적 관리, 사용자 정보 관리</td></tr>
<tr><td>소작업별 개발 기간</td><td>
<table>
<tr><td>소작업</td><td>개발 기간</td></tr>
<tr><td>사용자 정보 관리</td><td>1개월(1. 1 ~ 1. 31)</td></tr>
<tr><td>교과 과정 관리</td><td>1개월(2. 1 ~ 2. 28)</td></tr>
<tr><td>수업 관리</td><td>1개월(3. 1 ~ 3. 31)</td></tr>
<tr><td>수강 관리</td><td>1개월(4. 1 ~ 4. 30)</td></tr>
<tr><td>성적 관리</td><td>1개월(5. 1 ~ 5. 31)</td></tr>
<tr><td>테스트</td><td>1개월(6. 1 ~ 6. 30)</td></tr>
</table>
</td></tr>
<tr><td>개발 순서</td><td>사용자 정보 관리 → 교과 과정 관리 → 수업 관리 → 수강 관리 → 성적 관리</td></tr>
<tr><td>필요 자원</td><td>개발 툴(파워빌더), 개발용 PC, 개발 공간</td></tr>
</table>

08. 일정 계획

■ 일정 계획의 시작: 작업 분할 구조도(WBS)

▪ WBS

- 프로젝트 목표를 달성하는 데 필요한 활동과 업무를 세분화하는 작업
- 작업 패키지: 계층 구조에서 최하위에 있는 항목, 해당 업무의 담당자를 할당할 수 있을 정도로 작게 나눔

• WBS의 용도와 장점

- 사용자와 개발자의 의사소통 도구로 사용
- 프로젝트 업무 내역을 가시화할 수 있어 관리가 용이
- 프로젝트 팀원의 책임과 역할이 분명
- 필요 인력과 일정 계획을 세우는 데 기초로 활용
- 개발비 산정 시 기초로 활용
- 성과 측정 및 조정 시 기준선으로 활용할 수 있음

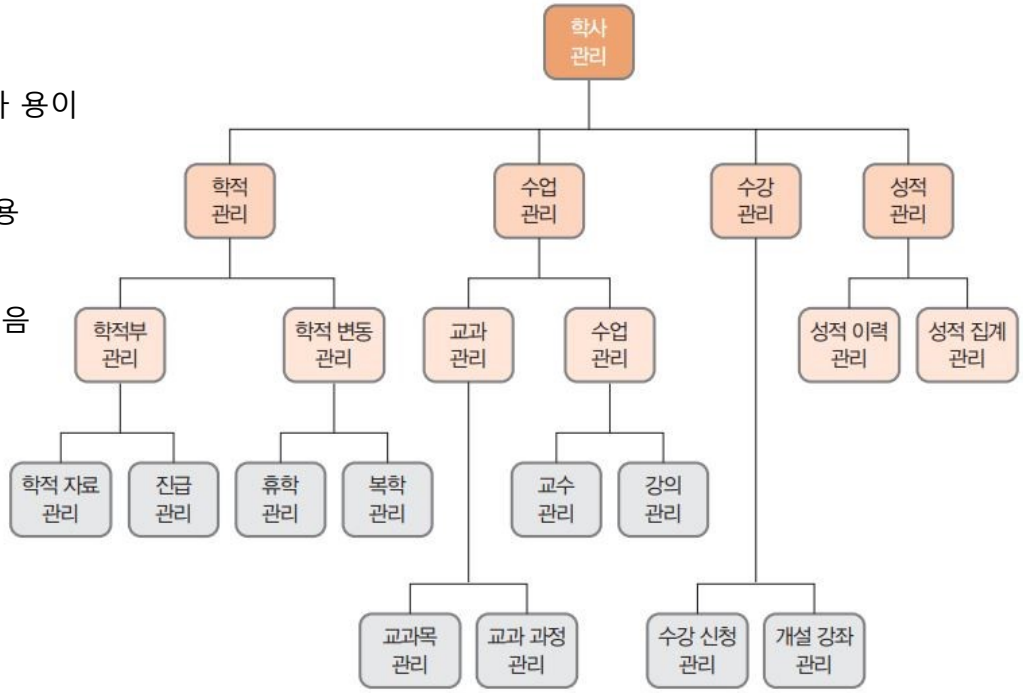
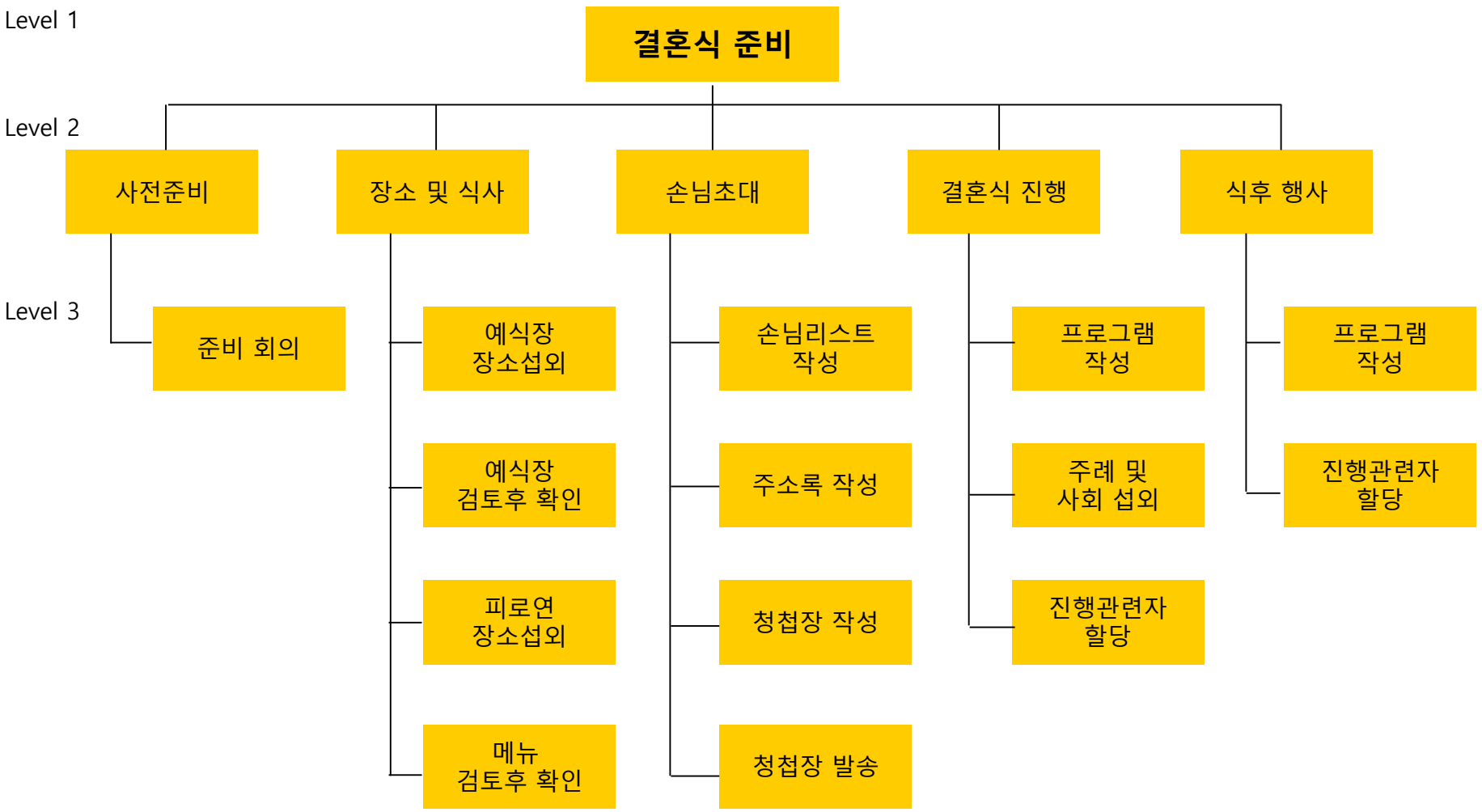
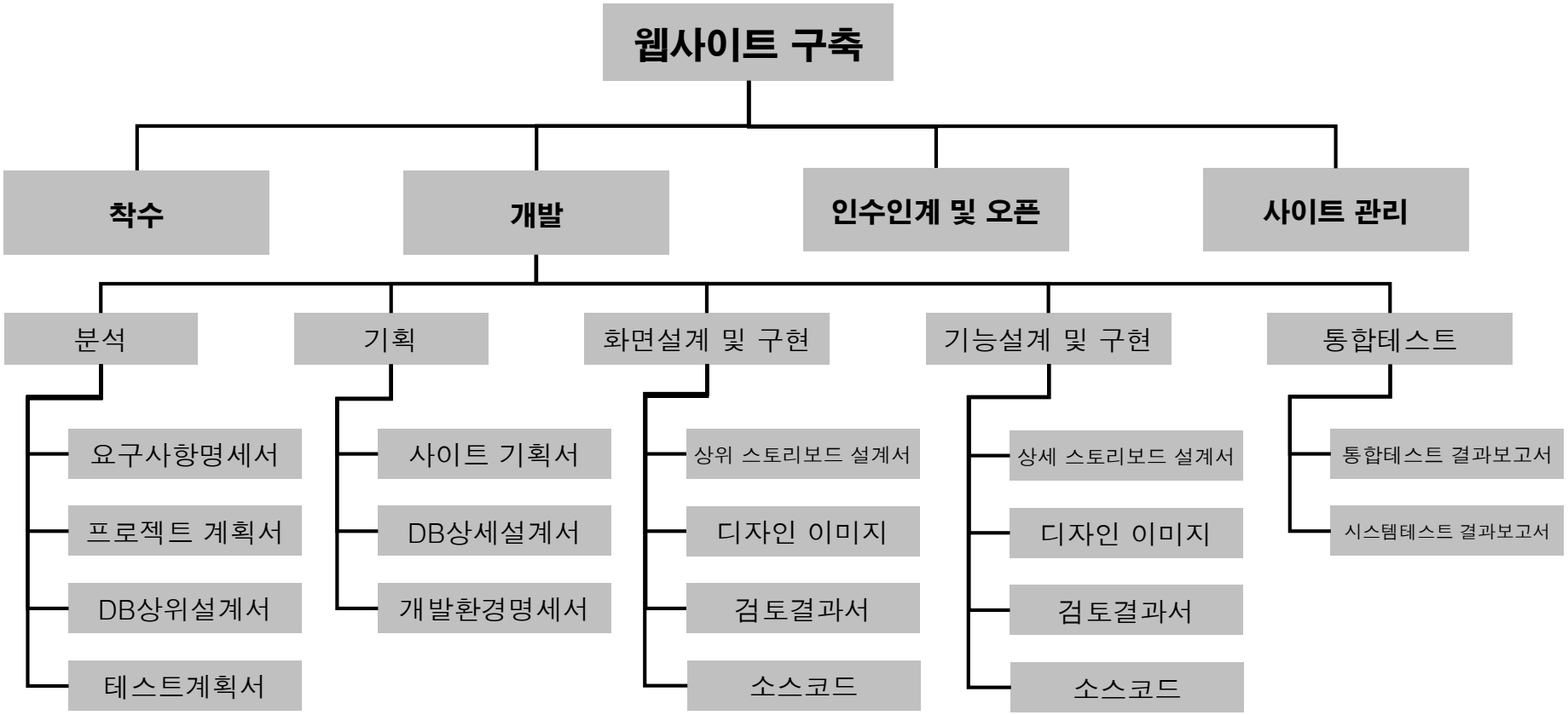


그림 3-12 학사관리시스템의 WBS

WBS 작성 예제


Level 1
Level 2
Level 3
결혼식 준비
사전준비
장소 및 식사
손님초대
결혼식 진행
식후 행사
준비 회의
예식장 장소섭외
예식장 검토후 확인
피로연 장소섭외
메뉴 검토후 확인
손님리스트 작성
주소록 작성
청첩장 작성
청첩장 발송
프로그램 작성
주례 및 사회 섭외
진행관련자 할당
프로그램 작성
진행관련자 할당

WBS 작성 예제


웹사이트 구축
착수
개발
인수인계 및 오픈
사이트 관리
분석
기획
화면설계 및 구현
기능설계 및 구현
통합테스트
요구사항명세서
프로젝트 계획서
DB상위설계서
테스트계획서
사이트 기획서
DB상세설계서
개발환경명세서
상위 스토리보드 설계서
디자인 이미지
검토결과서
소스코드
상세 스토리보드 설계서
디자인 이미지
검토결과서
소스코드
통합테스트 결과보고서
시스템테스트 결과보고서

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ 네트워크 차트(PERT/CPM)

- 네트워크 차트의 개요
  - WBS의 작업 순서, 소요 기간 등을 네트워크 형태의 그래프로 표현한 후 일의 중요도와 일정 관리를 명확히 하는 데 사용
  - 전체 작업 일정을 세분화해 지연을 사전에 예방할 수 있고, 개발 기간 단축에 활용해 일정을 효율적으로 관리할 수 있음
  - 유사성과 상호 장단점 때문에 PERT/CPM이라 해 두 기법을 혼합한 방법을 사용하기도 함

- PERT(퍼트)
  - 프로그램을 평가하고 검토하는 프로젝트 관리 기법
  - 프로젝트 진행 상황을 통계적인 방법으로 파악하고 이를 통해 일정 계획 및 통제를 할 수 있도록 고안

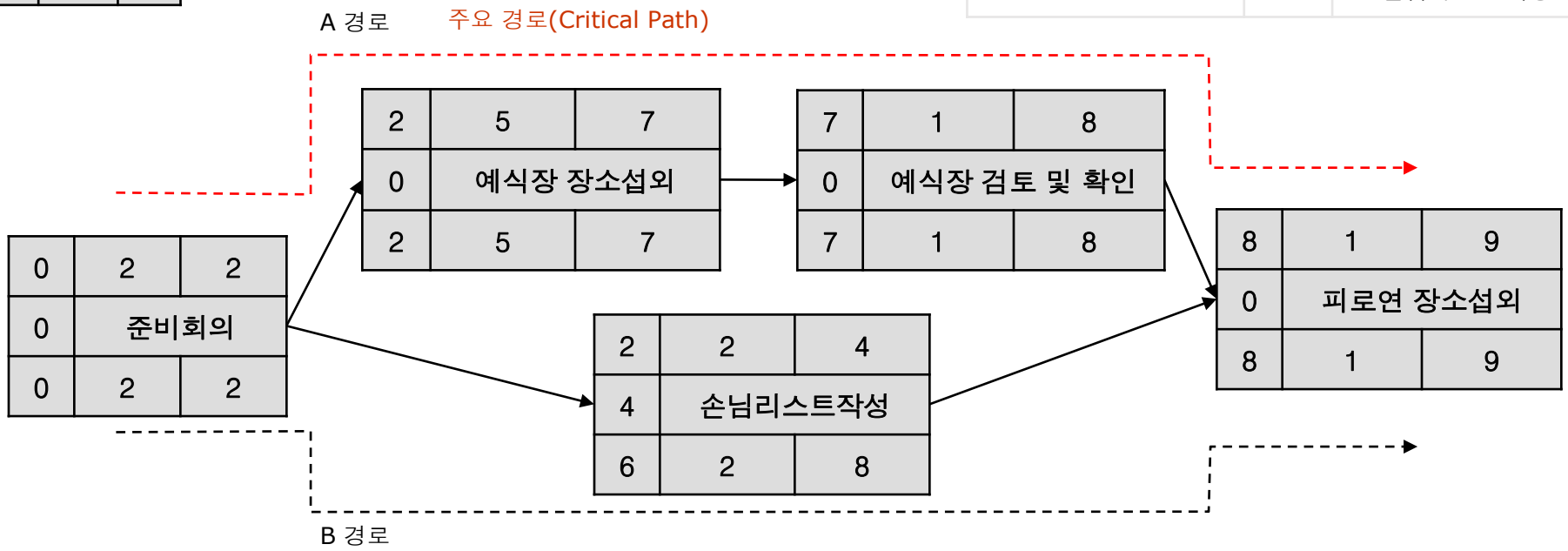
- CPM
  - 미 듀퐁사에서 화학 처리 공장의 건설 계획을 조직적으로 추진하기 위해 개발
  - 건설 공사와 같이 단위 작업이 확정적 소요 시간을 갖는 프로젝트인 경우에 적합

퍼트 차트(PERT Chart)

ES(Earliest Start Time): 해당 단위 작업이 가장 빨리 시작할 수 있는 일자
EF(Earliest Finish Tme): 해당 단위 작업이 가장 빨리 종료할 수 있는 일자
LS(Latest Start): 해당 단위 작업이 가장 늦게 시작하는 일자
LF(Latest Finish): 해당 단위 작업이 가장 늦게 종료할 수 있는 일자
Du(Duration): ES에서 EF까지, LS에서 LF까지의 작업 기간
FT(Float Time): LS에서 ES 사이의 여유 기간
주요경로 / 임계경로 : 임계 경로: 여유 시간이 없는 경로

단위 작업	기간	선행작업
준비회의	2	-
예식장 장소섭외	5	준비회의
예식장 검토/확인	1	예식장 장소섭외
손님리스트 작성	2	준비회의
피로연 장소섭외	1	예식장 검토/확인, 손님리스트 작성

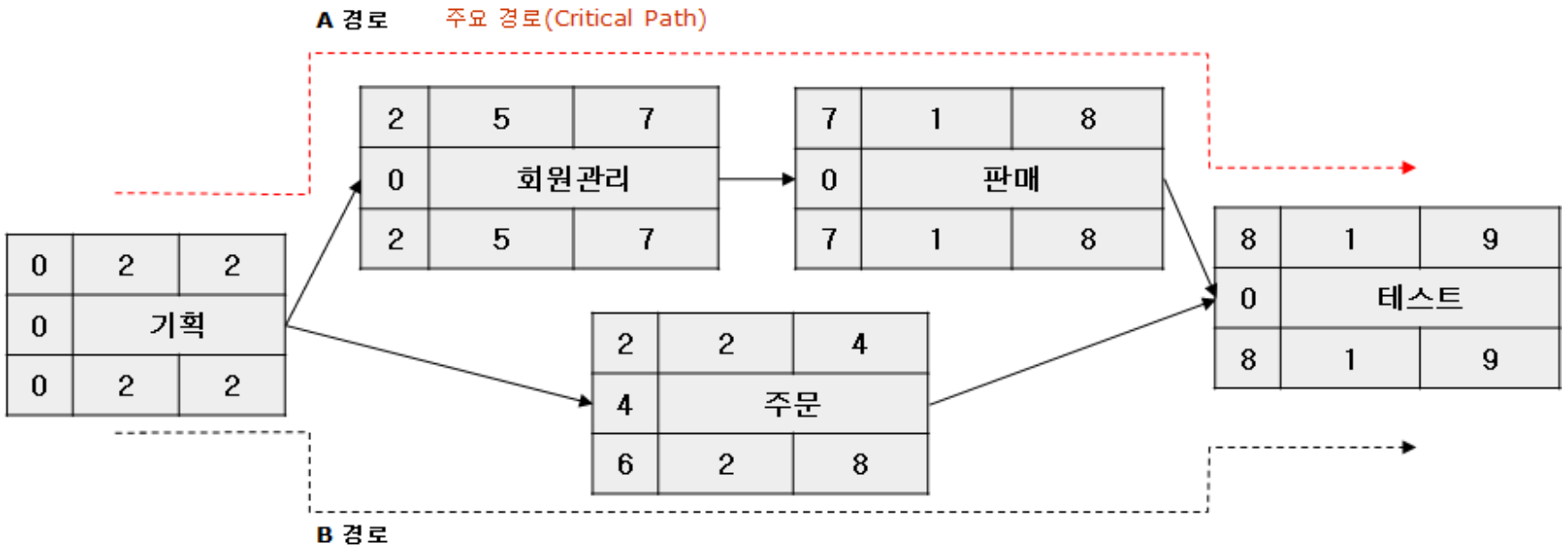
ES	Du	EF
FT	단위작업	
LS	Du	LF



ES(Earliest Start Time): 해당 단위 작업이 가장 빨리 시작할 수 있는 일자
EF(Earliest Finish Tme): 해당 단위 작업이 가장 빨리 종료할 수 있는 일자
LS(Latest Start): 해당 단위 작업이 가장 늦게 시작하는 일자
LF(Latest Finish): 해당 단위 작업이 가장 늦게 종료할 수 있는 일자
Du(Duration): ES에서 EF까지, LS에서 LF까지의 작업 기간
FT(Float Time): LS에서 ES 사이의 여유 기간
주요경로 / 임계경로 : 임계 경로: 여유 시간이 없는 경로

ES	Du	EF
FT	단위작업	
LS	Du	LF

단위 작업	기간	선행작업
기획	2	-
회원관리	5	기획
판매	1	회원관리
주문	2	기획
테스트	1	판매, 주문



08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

- CPM으로 네트워크를 그리려면 학사관리 애플리케이션을 수행하는 데 필요한 작업, 선행 작업, 작업의 소요 기간이 필요

표 3-18 CPM 네트워크를 위한 활동 목록

작업	작업 설명	선행 작업	소요 기간(주)
A	개인 정보 등록/수정/조회/삭제 프로그램 개발	–	2
B	학적 변동 자료 등록/수정/조회/삭제 프로그램 개발	A	6
C	휴 · 복학 및 자퇴 등록/수정/조회/삭제 프로그램 개발	A	4
D	교과 과정 등록/수정/조회/삭제 프로그램 개발	B, C	2
E	유사/동일 과목 등록/수정/조회/삭제 프로그램 개발	D	4
F	개설 강좌 등록/수정/조회/삭제 프로그램 개발	D	3
G	수강 과목 등록/수정/조회/삭제 프로그램 개발	E	2
H	시간표 등록/수정/조회/삭제 프로그램 개발	E	4
I	성적 등록/수정/조회/삭제 프로그램 개발	F	2
J	장학생 등록/수정/조회/삭제 프로그램 개발	F	1
K	등록금 등록/수정/조회/삭제 프로그램 개발	G, H	2
L	졸업 사정 등록/수정/조회/삭제 프로그램 개발	I, K	2
M	사회봉사 실적 등록/수정/조회/삭제 프로그램 개발	J, L	3

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

- CPM 작업 과정

① CPM 네트워크를 그림

- 노드와 간선을 이용해 초기의 CPM 네트워크를 그림
- 노드는 작업을, 간선은 작업들 간의 선후 의존 관계를 나타냄

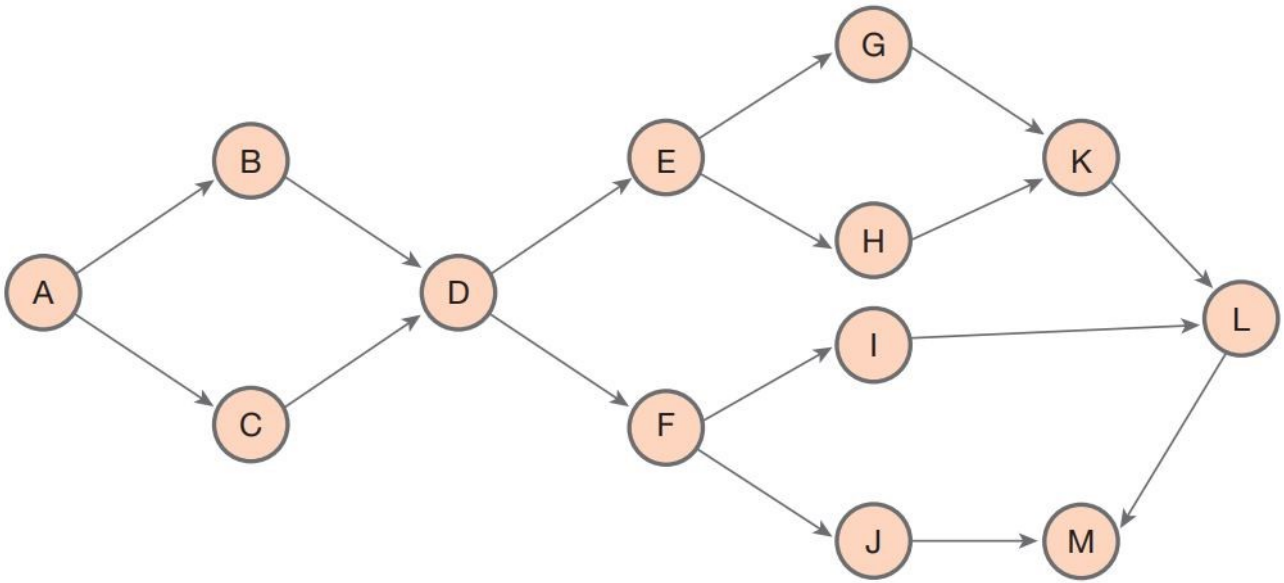


그림 3-13 CPM 네트워크

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

② ES값을 구함

- ES: 가능한 빨리 시작할 수 있는 시간으로, 선행 작업이 완료되었을 때 해당 작업을 시작할 수 있는 가장 빠른 시점
- ES 값을 구할 때는 맨 앞(작업 A)에서 끝 방향으로 가며 계산
- ES에서 주의할 부분은 두 작업이 합류하는 지점의 작업 시작 시간은 두 작업이 모두 완전히 끝났을 때

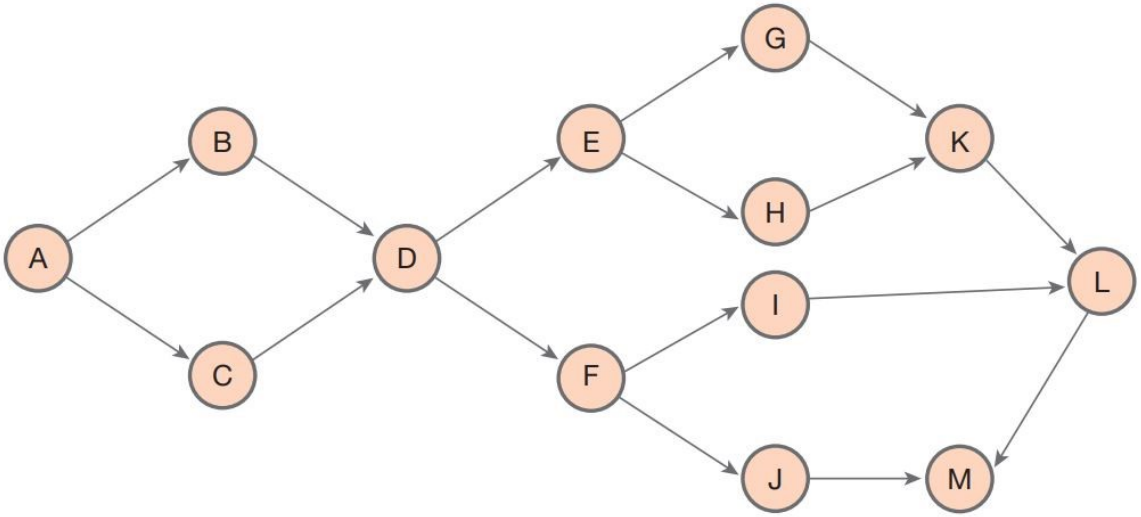


표 3-19 작업별 가장 빨리 시작할 수 있는 시간 (단위: 주)

작업	A	B	C	D	E	F	G	H	I	J	K	L	M
작업 시작 시간	0	2	2	8	10	10	14	14	13	13	18	20	22
작업 시간	2	6	4	2	4	3	2	4	2	1	2	2	3

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

- CPM 작업 과정

③ EF값을 구함

- EF: 가장 빠른 시작 시간(ES)으로 시작했을 때의 가장 빠른 완료 시간(ES+작업 소요 시간)

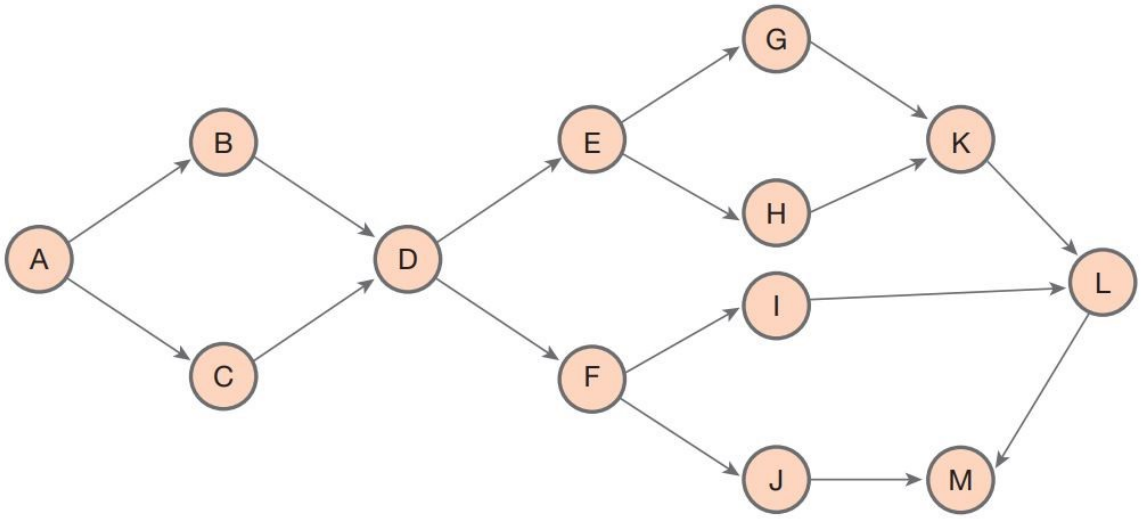


그림 3-13 CPM 네트워크

표 3-20 작업별 가장 빨리 완료할 수 있는 시간 (단위: 주)

작업	A	B	C	D	E	F	G	H	I	J	K	L	M
EF	2	8	6	10	14	13	16	18	15	14	20	22	25

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

④ LS값을 구함

- LS: 어떤 작업을 늦어도 시작해야 하는 시간, 즉 가장 늦게 시작할 수 있는 시간
- 맨 뒤(작업 M)에서 앞(작업 A) 방향으로 계산
- 두 지점으로 갈라지는 곳에서는 다음 작업의 시작 시간에 영향을 주지 않고 시작할 수 있는 시간을 찾아 기입

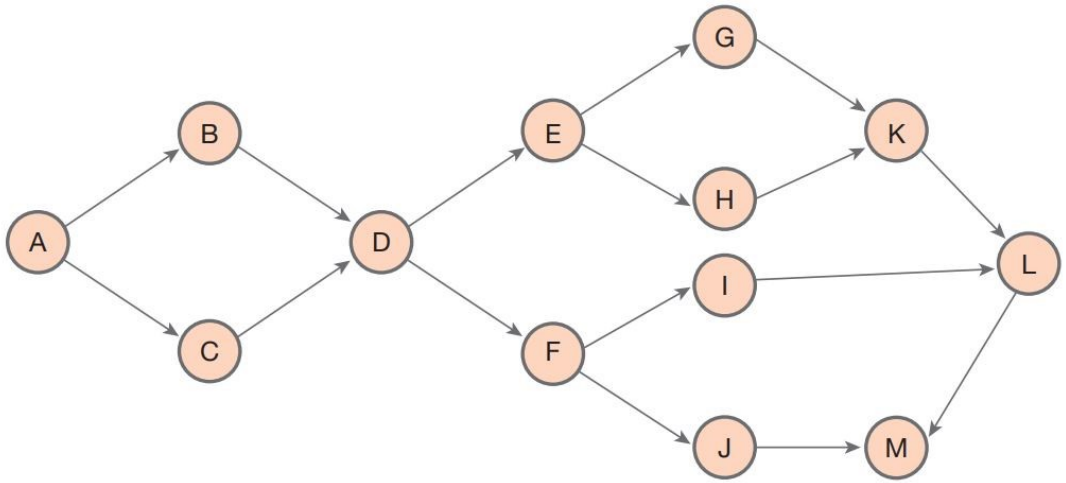


그림 3-13 CPM 네트워크

표 3-21 작업별 가장 늦게 시작할 수 있는 시간 (단위: 주)

작업	A	B	C	D	E	F	G	H	I	J	K	L	M
작업 시작 시간	0	2	4	8	10	15	16	14	18	21	18	20	22
작업 시간	2	6	4	2	4	3	2	4	2	1	2	2	3

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

⑤ LF값을 구함

- LF: 가장 늦게 시작할 수 있는 시간(LF)에 시작해 작업을 완료한 시간(LS+작업 소요 시간)

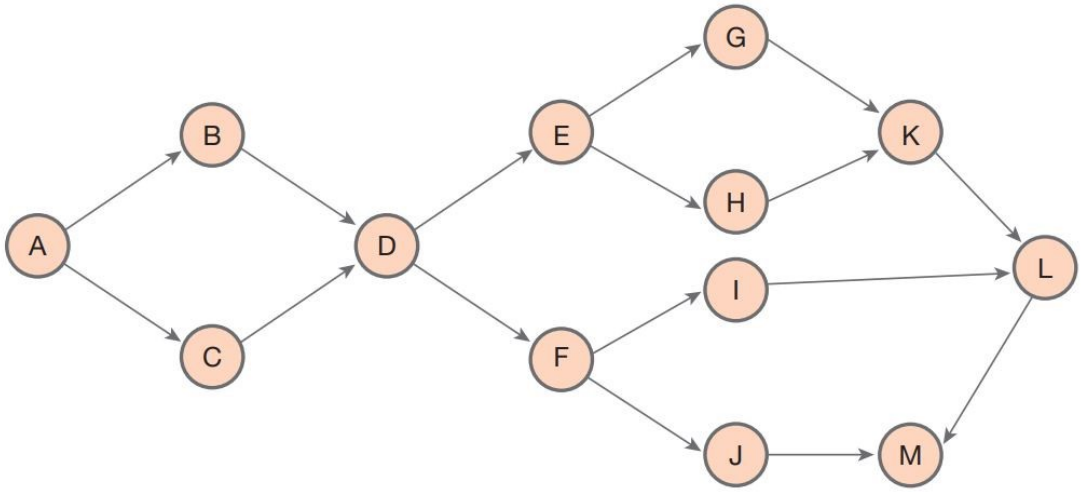


그림 3-13 CPM 네트워크

표 3-22 작업별 가장 늦게 시작했을 때 완료할 수 있는 시간 (단위: 주)

작업	A	B	C	D	E	F	G	H	I	J	K	L	M
작업 시작 시간	0	2	4	8	10	15	16	14	18	21	18	20	22
작업 시간	2	6	4	2	4	3	2	4	2	1	2	2	3
작업 완료 시간	2	8	8	10	14	18	18	18	20	22	20	22	25

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

⑥ ST값을 구함

- ST: 여유 시간, 각 작업에서 '가장 늦게 시작하는 시간'에서 '가장 빨리 시작하는 시간'을 빼면 구할 수 있음
- 전체 작업 시간을 줄이고 싶을 때는 여유 시간이 존재하는 작업의 시간을 줄이면 됨

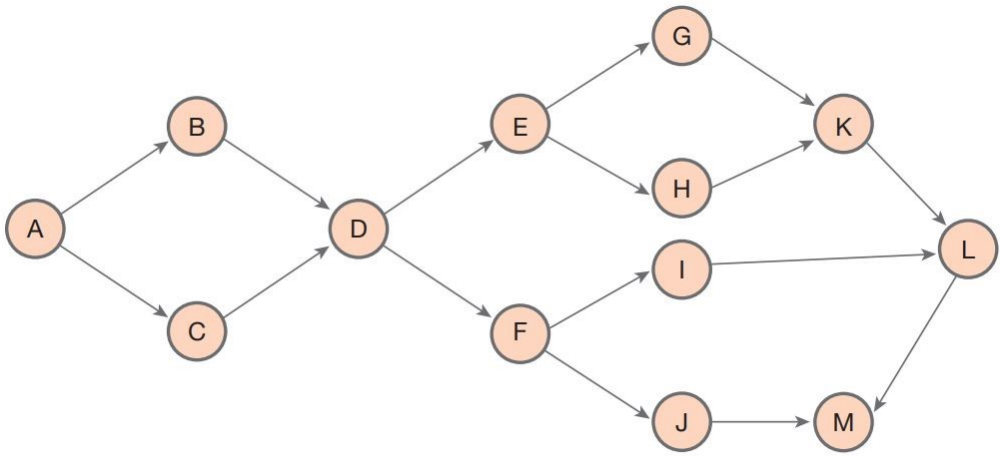


그림 3-13 CPM 네트워크

표 3-23 여유 시간 (단위: 주)

작업	A	B	C	D	E	F	G	H	I	J	K	L	M
작업별 빠른 시작 시간	0	2	2	8	10	10	14	14	13	13	18	20	22
작업별 늦은 시작 시간	0	2	4	8	10	15	16	14	18	21	18	20	22
여유 시간	0	0	2	0	0	5	2	0	5	8	0	0	0

08. 일정 계획

■ 일정 계획 기법 1: 네트워크 차트

▪ CPM 작업 과정

⑦ 임계 경로를 구함

- 임계 경로: 여유 시간이 없는 경로(예시에서는 경로 'A-B-D-E-H-K-L-M'가 임계 경로)
- 임계 경로에는 여유 시간이 없으므로 모든 일정 계획은 임계 경로에 좌우됨
- 임계 경로에서 벗어난 활동을 수행하는 데 걸리는 시간이 한도 내에서 늦춰지거나 당겨 질 경우에는 전체 프로젝트 완료 시간에 변화가 없음

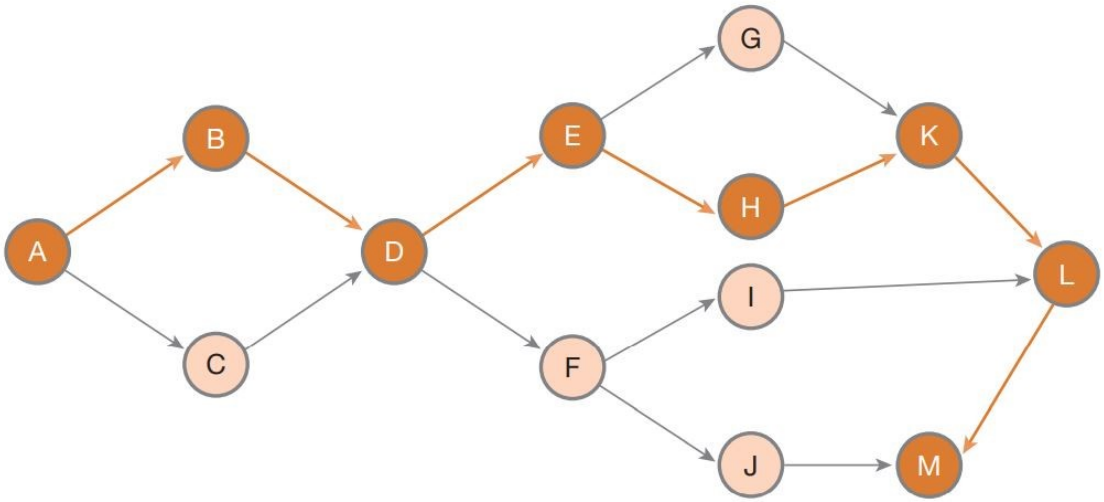


그림 3-14 CPM 네트워크와 임계 경로

08. 일정 계획

■ 일정 계획 기법 2: 간트 차트를 이용한 일정표 작성

▪ 간트 차트

- 프로젝트 일정 관리를 위한 바 형태의 도구
- 프로젝트의 주요 활동을 파악한 후, 각 활동의 일정을 시작하는 시점과 끝나는 시점을 연결한 막대 모양으로 표시
- 전체 일정을 한눈에 볼 수 있음

ID	활동명	시작일	완료일	기간
1	학적부 관리	21.01.01	21.02.28	2
2	학적 변동 관리	21.01.01	21.01.31	1
3	교과 관리	21.03.01	21.05.31	3
4	수업 관리	21.06.01	21.07.31	2
5	수강관리	21.07.01	21.09.30	3
6	성적 관리	21.10.01	21.12.31	3

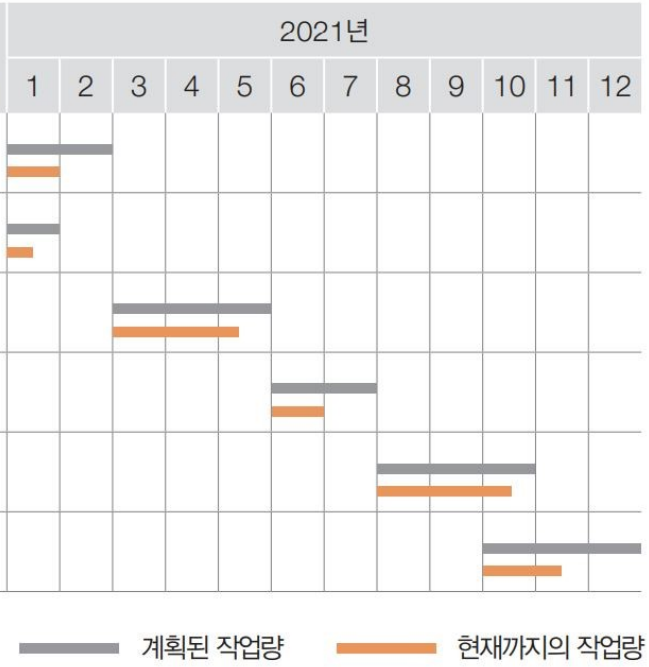


그림 3-15 간트 차트를 이용한 프로젝트 일정 계획표

09. 위험 분석

■ 위험 분석의 이해

위험 : 일정, 비용, 범위, 품질과 같은 프로젝트 목적들 중 적어도 하나의 목적에 나쁜 영향을 끼칠 지도 모르는 이벤트 혹은 상태

- 예시) 겨울철 눈에 대비
  - 위험 예방
    - 내일 자동차로 출근하거나 여행을 계획했다면 오늘 일기예보를 듣고 교통수단을 결정
    - 여행 계획을 세웠다면 아쉽지만 며칠 미루는 것

그림 3-16 위험 예방

09. 위험 분석

■ 위험 분석의 이해

- 예시) 겨울철 눈에 대비
  - 도구를 사용한 위험 예방
    - 폭설이 예상되지만 자동차로 출근할 수밖에 없거나, 모처럼 계획한 가족 여행을 미루기 어려울 경우가 있음
    - 스노타이어나 체인 등을 사용해 위험을 최소화

그림 3-17 도구를 사용한 위험 예방

09. 위험 분석

■ 위험 분석의 이해

- 예시) 겨울철 눈에 대비
  - 위험 예방을 위한 도구 준비
    - 여행을 떠날 때는 눈이 오지 않았지만 자동차 여행 중에 갑자기 폭설을 만날 수도 있음
    - 비상사태에 대비해 자동차 트렁크에 체인을 넣어 두거나 미끄럼을 방지할 수 있는 타이어 스프레이를 준비

그림 3-18 위험 예방을 위한 도구 준비

09. 위험 분석

■ 위험 관리 절차

1. 리스크 식별 : 프로젝트의 성공에 위협이 되는 불안 요소들을 파악하는 활동.(프로젝트 수행 중 대응책을 마련해 놓을 수 있음.)
2. 리스크 계량화 : 식별된 리스크 요소들의 발생 확률과 발생 시 프로젝트에 미치는 영향 정도 구분 작업.(발생 확률과 영향도는 과거 데이터와 프로젝트 참여 팀원들의 경험에 근거하여 작성.)
3. 리스크 우선순위 선정 : 리스크 우선순위 설정 이유는 프로젝트에 제공되는 인력, 기간, 예산이 제한되어 있어 모든 리스크에 대응하기 충분하지 않기 때문.
4. 리스크 관리 계획 : 리스크 발생 시 영향을 최소화 할 수 있도록 리스크 관리 계획을 작성. 리스크 관리 계획에는 리스크를 언제, 어떠한 방법으로 관리할 것인가 대한 내용 등이 포함
5. 위험 감시 및 리스크 해결 : 리스크는 프로젝트의 어느 시점에서나 문제로 발생할 수 있으므로 리스크 관리는 프로젝트 초반부터 프로젝트 종료 시까지 수행. 프로젝트 수행 중 주기 별로 식별된 리스크 요소에 대한 평가 수행 (주간 회의나 마일스톤 회의에서 리스크 요소의 발생 가능성을 판단)
6. 결과 측정 및 문서화 : 리스크 관리 계획에 따른 대응방안의 수행 결과가 어느 정도 효과가 있었는지에 대해 평가하고, 그 결과를 기록. 리스크 관리 결과는 조직의 리스크 요소 리스트에 다시 반영 시킨다.(다른 유사한 프로젝트에서 같은 실수를 반복하지 않을 수 있고, 리스크 요소에 대한 보다 효과적인 대응 방안을 마련할 수 있음.)

1. 리스크 식별

- 이전 유사 프로젝트를 참고하여 위험 요소 식별
- 브레인스토밍 하여 위험 요소 식별

- 리스크 요소와 원인 예

분야	리스크 요소	원인
고객	고객의 요구사항의 변경	개발되는 제품에 대한 파악 부족
	고객의 재정적 어려움	고객에 대한 조사 부족
인력	인력의 수를 잘못 예상	산정 방식의 문제점, 요구사항의 잘못된 파악
	개발 도중에 인력이 그만둠	대화 부족, 인력에 대한 정보부족
	새로운 인력으로 인한 팀워크의 불안	준비작업 부족
기술	개발 언어에 대한 불확실	교육의 부족
	주요 기술의 부족한 파악	개발 분야에 대한 분석 부족
	예상 보다 적은 재사용	개발 분야에 대한 분석 부족
개발 프로세스	개발기간의 부족	요구사항의 잘못된 파악, 인력 및 기술의 부족
	비용 부족	요구사항의 잘못된 파악, 산정 방식의 문제점
제품 기능	예상 보다 많은 사용자	요구사항의 잘못된 파악
	개발 영역에 대한 잘못된 파악	요구사항의 잘못된 파악

2. 리스크 계량화

- **리스크 요소들을 실제 발생할 수 있는 확률과 발생 시 프로젝트에 미치는 영향의 정도에 따라 구분**
- **각 리스크 요소 별 발생 확률과 영향도는 과거 프로젝트의 데이터와 리스크 계량화에 참여한 프로젝트 팀원의 경험에 근거하여 작성**
  - 각 요소 별 리스크 발생 확률 예

발생 확률	발생확률 범위(%)
상	80% 이상
중	30% ~ 80%
하	30% 이하

  - 리스크 요소 영향도의 예

영향도	비용/일정 초과 범위
상	20%이상
중	5% ~ 20%
하	5% 이하

3. 리스크 우선순위 선정

■ 리스크 우선순위를 설정하여 대응책을 만들고 리스크를 관리

- 프로젝트에 제공되는 인력, 기간, 예산의 제한으로 인하여 모든 리스크 요소에 대응하기에는 시간이 충분하지 않기 때문

- 리스크 우선순위 기준의 예

영향도 / 발생확률	상	중	하
상	1	2	3
중	2	3	4
하	3	4	5

리스크 요소 별 우선순위의 예

분야	리스크 요소	원인	발생확률	영향도	우선순위
고객	고객의 요구사항의 변경	개발되는 제품에 대한 파악 부족	상	상	1
	고객의 재정적 어려움	고객에 대한 조사 부족	하	중	4
인력	인력의 수를 잘못 예상	산정 방식의 문제점, 요구사항의 잘못된 파악	중	상	2
	개발 도중에 인력이 그만둠	대화 부족, 인력에 대한 정보부족	상	상	1
	새로운 인력으로 인한 팀워크의 불안	준비작업 부족	중	중	3
기술	개발 언어에 대한 불확실	교육의 부족	하	중	4
	주요 기술의 부족한 파악	개발 분야에 대한 분석 부족	중	상	3
	예상 보다 적은 재사용	개발 분야에 대한 분석 부족	하	하	5
개발 프로세스	개발기간의 부족	요구사항의 잘못된 파악, 인력 및 기술의 부족	상	상	1
	비용 부족	요구사항의 잘못된 파악, 산정 방식의 문제점	상	상	1
제품 기능	예상 보다 많은 사용자	요구사항의 잘못된 파악	하	상	5
	개발 영역에 대한 잘못된 파악	요구사항의 잘못된 파악	상	상	1

4. 리스크 관리 계획

- 식별된 리스크 요소에 대해서는 리스크의 영향을 최소화 할 수 있도록 리스크 관리 계획을 작성이 필요
- 리스크 관리 계획에는 리스크를 언제, 어떠한 방법으로 관리할 것인가 대한 내용 등이 포함

분야	리스크 요소	대응 방안
고객	고객의 요구사항의 변경	요구사항에 대한 고객과 재협의
	고객의 재정적 어려움	개발 범위의 단축
인력	인력의 수를 잘못 예상	예비 인력의 투입
	개발 도중에 인력이 그만둠	예비 인력의 투입
	새로운 인력으로 인한 팀워크의 불안	커뮤니케이션 방법의 추가
기술	개발 언어에 대한 불확실	필요 교육의 실시
	주요 기술의 부족한 파악	기술에 대한 교육 및 컨설팅 실시
	예상 보다 적은 재사용	기술에 대한 교육 및 컨설팅 실시
개발 프로세스	개발기간의 부족	요구사항의 재분석, 예비 인력의 투입
	비용 부족	개발 비용의 재 산정 및 재 편성
제품 기능	예상 보다 많은 사용자	요구사항에 대한 고객과 재협의
	개발 영역에 대한 잘못된 파악	요구사항에 대한 고객과 재협의

5. 위험 감시 및 리스크 해결

- **리스크는 프로젝트의 어느 시점에서나 문제로 발생할 수 있으므로 리스크 관리는 프로젝트 초반부터 프로젝트 종료 시까지 수행**
- **프로젝트 수행 중 주기 별로 식별된 리스크 요소에 대한 평가 수행**
  - 예) 주간 회의나 마일스톤 회의에서 리스크 요소의 발생 가능성을 판단

6. 결과 측정 및 문서화

- **리스크 관리 계획에 따른 대응방안의 수행 결과가 어느 정도 효과가 있었는지에 대해 평가하고, 그 결과를 기록해야 함**
- **리스크 관리 결과는 조직의 리스크 요소 리스트에 다시 반영 시켜야 함**
  - 다른 유사한 프로젝트에서 같은 실수를 반복하지 않을 수 있고, 리스크 요소에 대한 보다 효과적인 대응 방안을 마련할 수 있음