

# 5. 프로젝트 계획 및 통제

---

# 주요내용

---

- ❖ 프로젝트 계획서란?
- ❖ 프로젝트 팀 구성은 어떻게 해야 하는가?
- ❖ WBS란?
- ❖ 프로젝트 산정 기법은 어떤 것들이 있는가?
- ❖ 일정 계획 방식은 어떤 것들이 있는가?
- ❖ 위험 관리란?

# 목차

---

## ❖ 강의 내용

- 프로젝트 계획서
- 프로젝트 팀 구성
- WBS
- 프로젝트 산정 기법
- 일정 계획 방식
- 위험 관리

## ❖ 팀 프로젝트 (6, 7, 9주차)

- 요구사항 명세서 작성 및 제출
- 프로젝트 계획서 작성 및 제출

# 프로젝트 계획서

---

## ❖ 의미

- 프로젝트 관리자 뿐만 아니라 프로젝트 참여자 모두가 프로젝트를 진행해 가면서 참조하는 프로젝트의 중심이 되는 문서

## ❖ 작성 순서

- 프로젝트 관리자는,
  - 프로젝트 태스크 파악
  - 각 태스크를 수행하기 위해 필요한 노력 예측
  - 인적 자원 및 기타 자원을 각 태스크에 할당
  - 일정 계획 작성
- 프로젝트 참여자의 검토를 거쳐 합의 하에 프로젝트 채택함

# IEEE 1058.1-1987 프로젝트 계획서 양식

---

## 1개요

- 1.1 프로젝트 개요
- 1.2 프로젝트 산출물
- 1.3 계획서의 변경기록
- 1.4 참고문헌
- 1.5 정의와 약어

## 2프로젝트 조직

- 2.1 프로세스 모델
- 2.2 조직 구조
- 2.3 조직의 범위와 인터페이스
- 2.4 프로젝트 책임

## 3관리적 프로세스

- 3.1 관리적 목적과 우선순위
- 3.2 가정과 제한
- 3.3 위험관리
- 3.4 통제 메커니즘
- 3.5 인력

## 4기술적 프로세스

- 4.1 방법론 도구
- 4.2 소프트웨어 문서화
- 4.3 지원기능

## 5작업, 스케줄, 예산

- 5.1 작업
- 5.2 작업간 의존관계
- 5.3 자원요구
- 5.4 예산 및 자원할당
- 5.5 스케줄

# 프로젝트 계획서의 역할 및 중요성

---

- ❖ **프로젝트 진행 과정의 주기적 통제의 기본**
  - 주간, 월간 회의를 통해 점검
  
- ❖ **프로젝트가 크고 참여자가 많을수록 잘 짜여진 프로젝트 계획서가 중요함**
  - 프로젝트 계획서가 현실적으로 작성되어 전체 프로젝트 진행상황 파악에 크게 문제가 되지 않아야 함

# 프로젝트 팀 구성

---

# 프로젝트 팀 구성

---

## ❖ 팀 구성의 기준

- 프로젝트 기간과 크기

## ❖ 팀 구성원의 역할

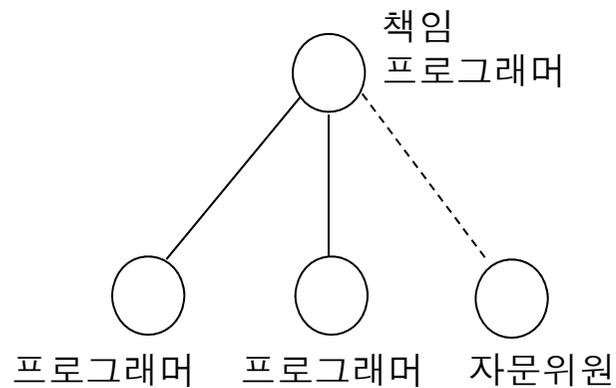
- 프로젝트 팀장
- 분석 및 설계자
- 개발자
- 품질 보증 담당자
- 산출물 관리 담당자
- 테스트 담당자
- ...

# 팀장과 구성원의 2단계 구조 (1/2)

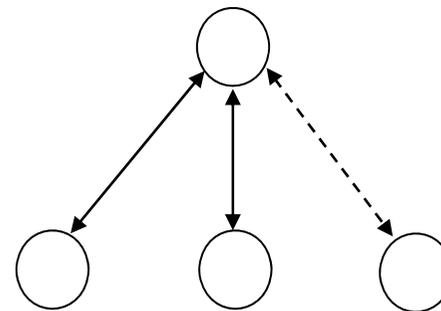
## ❖ 소개

- 프로젝트 책임자인 팀장은 상위 단계에, 나머지 참여자는 전부 다음 단계에 속함
- 일반적인 소규모 프로젝트가 가장 많이 취하고 있는 팀 구조

## ❖ 구조



(a) 조직 구성



(b) 의사 소통 경로

# 팀장과 구성원의 2단계 구조 (2/2)

---

## ❖ 역할 소개

- **책임 프로그래머**
  - 팀의 운영에 대한 결정권한 및 운영에 대한 책임을 가짐
- **팀원**
  - 프로젝트 수행 중 팀장에게 보고하고 지시를 받음
- **팀장**
  - 프로젝트 계획을 작성하고 통제함

## ❖ 단점

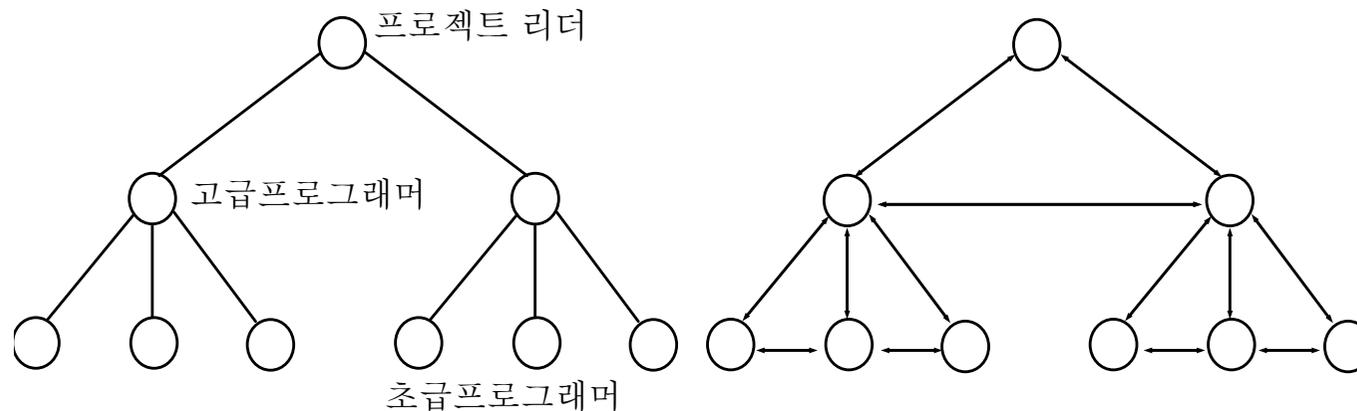
- **팀장 중심의 팀 구성으로 팀장의 능력에 따라 프로젝트 성패가 좌우될 가능성이 크다.**

# 계층적 팀 구성 (1/2)

## ❖ 소개

- 팀의 구성이 둘 이상의 단계로 나누어짐
- 프로젝트가 크고, 참여인원이 많을 때 많이 채택되는 방식

## ❖ 구조



[a] 조직 구성

[b] 의사 소통 경로

# 계층적 팀 구성 (2/2)

---

## ❖ 역할 소개

### - 각 그룹의 장(리더)

- 그룹원들을 책임지고 관리

### - 팀장

- 그룹 리더들로부터 보고를 받고, 그룹 리더들과 의논하고 지시사항을 전달

## ❖ 장점

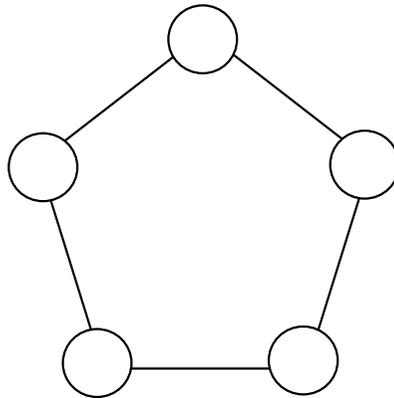
- 그룹원들은 그룹 리더들과, 그룹 리더들은 팀장과 의사소통을 하기 때문에 의사 교환 경로를 줄일 수 있음

# 민주적 팀 구성

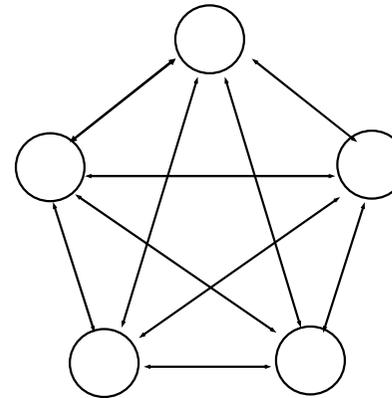
## ❖ 소개

- 모든 팀원이 리더의 역할을 하고, 중요한 의사 결정은 팀원 모두가 참여

## ❖ 구조



(a) 조직 구성



(b) 의사 소통 경로

## ❖ 장/단점

- 팀원의 사기와 작업 만족도를 높이고, 의사 결정시 많은 의견을 통한 결정을 할 수 있다는 장점
- 의사 교환 경로가 많아 의견의 합의점을 찾는데 시간이 걸릴 수 있음

# 스케줄링(Scheduling)

---

# 스케줄링이란?

---

## ❖ 의미

- 프로젝트의 완성을 위해 수행되어야 할 작업을 나열한 후 연관 관계와 순서에 따라 기간 별로 나타내는 것

## ❖ 스케줄링 방식

- WBS(Work Breakdown Structure)
  - 프로젝트 중 수행되어야 하는 작업들을 파악

# WBS

---

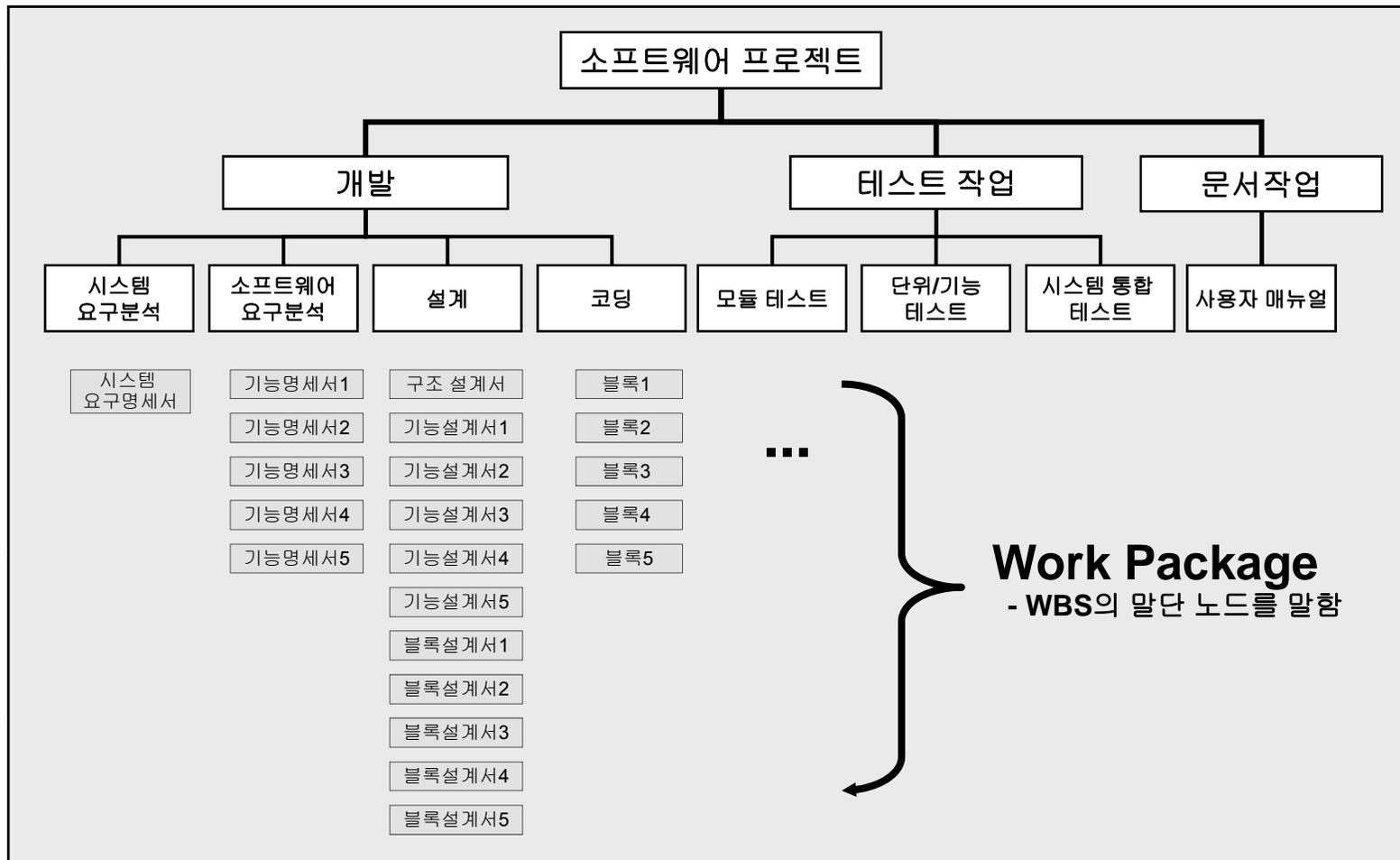
## ❖ WBS (Work Breakdown Structure)

- 프로젝트를 톱 다운(Top Down) 방식으로 세분화하여 프로젝트의 단위 작업에 대해 파악하는 기법

# WBS 작성 예제 – 결혼식 준비



# 폭포수 생명 주기 기반의 WBS 예제



# 프로젝트 산정

---

# 산정

---

## ❖ 개념

- 프로젝트 수행에 필요한 규모(Size), 공수(Effort), 비용(Cost) 등을 정량적으로 예측하는 것

# 산정의 방법

---

## ❖ 경험적 방법

- 프로젝트의 수량을 예측하기 위해서 노력과 시간에 대한 수식을 경험적으로 유도한 것
- 예: 델파이 기법

## ❖ 크기 중심 방법

- LOC(Line of Code : 프로그램 코드 라인 수)로 측정
- 예: LOC, COCOMO

## ❖ 기능 중심 방법

- 사용자 중심의 기능의 크기로 측정
- 예: 기능점수(Function Point)로 측정

# 델파이(Delphi) 기법(1/2)

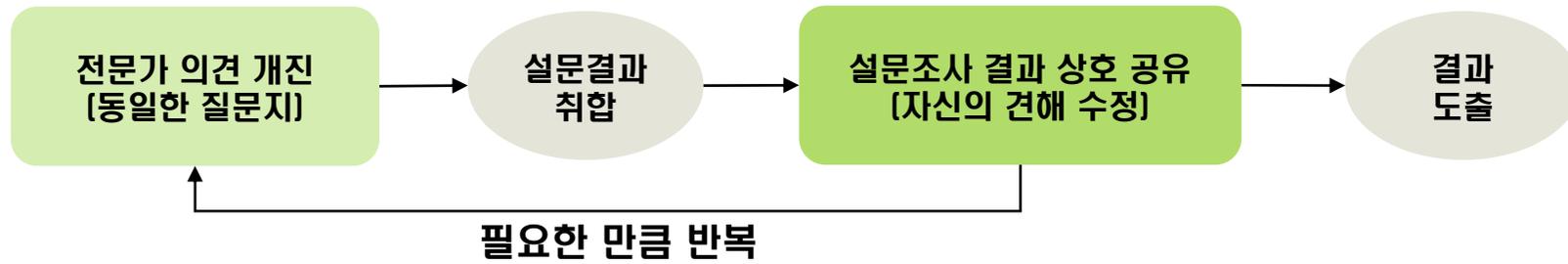
---

## ❖ 의미

- 경험적 산정 방법
- 다수의 전문가의 의견을 설문조사 등을 활용하여 수 회에 걸쳐 피드백(feedback)시켜 그들의 의견을 수렴하고 합의된 내용을 얻는 방법

# 델파이(Delphi) 기법(2/2)

## ❖ 산정 프로세스



# LOC(Lines Of Code)

---

## ❖ LOC에 대한 기대

- 윈도우 2000: 2900만 코드 라인
- 윈도우 2000, 윈도우 XP와 윈도우 서버 2003 제품군의 모든 버전 및 서비스 팩이 1억 라인 이상의 코드 라인...

## ❖ 의미

- 크기 중심적 산정 방법
- 프로그램 코드 라인의 수를 중심으로 산정

## ❖ 산정 프로세스

- 단계 1. 전체 프로그램을 모듈별로 분할
- 단계 2. 모듈별로 규모추정 및 규모 계산
  - $EV = (V_{opt} + 4 V_m + V_{pess}) / 6$
  - $V_{opt}$  = 낙관적 규모
  - $V_m$  = 보통의 규모
  - $V_{pess}$  = 비관적 규모
- 단계 3. 경험적 데이터를 이용한 개발 비용 및 개발 노력 추정
  - 생산성 (LOC / Man-Month)을 이용 => 노력 추정
  - LOC당 비용 ( $\alpha$  원/ LOC) => 비용 추정  
( $\alpha$ : 경험적 데이터로부터 산정함)

# LOC 산정의 예(1/2)

## ❖ 쇼핑몰 시스템: 모듈별로 분할한다.

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

## ❖ 각 모듈별로 규모 추정

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

# LOC 산정의 예(2/2)

---

## ❖ 경험적 데이터를 이용한 개발 노력 및 개발 비용 추정

### - 경험적 데이터

- 생산성: 620 LOC / Man-Month
- LOC당 비용: 3,000원 / LOC

### - 프로젝트 비용

- $1,882 \text{ LOC} \times 3,000 = 5,646,000\text{원}$

### - 개발 노력

- $1,882 \text{ LOC} / 620 \approx 3.0 \text{ M/M (Man-Month)}$

# COCOMO(Constructive Cost Model)

## ❖ 개념

- Boehm이 제시한 경험적 산정 모델
- 경험적으로 추출된 상수와 추정된 LOC를 기반으로 개발 노력과 개발 기간을 산정

## ❖ 수식

- $E = a \times (KLOC)^b$
- $D = c \times (E)^d$   
( E: 개발 노력, D: 개발기간, KLOC:Kilo Lines of Code(1000LOC))
- 프로젝트 유형별 상수값(상관 계수 테이블)

프로젝트	a	b	c	d
기본형	2.4	1.05	2.5	0.38
중간형	3.0	1.12	2.5	0.35
내장형	3.6	1.20	2.5	0.32

### •기본형

- 비교적 작고 간단한 프로젝트
- 소수의 경험이 있는 팀이 까다롭지 않은 요구사항을 갖고 프로젝트 수행

### 중간형

- 중간 정도의 크기와 복잡도를 가짐
- 다양한 경험을 가진 팀이 약간 까다로운 요구사항을 갖고 프로젝트 수행

### 내장형

- 제한된 하드웨어, 소프트웨어와 운영조건을 갖고 프로젝트 수행

# COCOMO 산정의 예

## ❖ 개발 프로젝트

- 쇼핑몰 시스템: 프로젝트 유형 중 “중간형”에 해당함
- 예상 규모: 2,103 LOC = 2.1 KLOC

## ❖ 산정

프로젝트	a	b	c	d
기본형	2.4	1.05	2.5	0.38
중간형	3.0	1.12	2.5	0.35
내장형	3.6	1.20	2.5	0.32

- $E = 3.0 \times (\text{KLOC})^{1.12} = 3.0 \times (2.1)^{1.12} \doteq 6.9 \text{ M/M (Man-Month)}$
- $D = 2.5 \times (E)^{0.35} = 2.5 \times (6.9)^{0.35} \doteq 4.9 \text{ M (Month)}$

# FP(Function Point) (1/5)

---

## ❖ 의미

- 소프트웨어를 구성하는 기능의 크기를 통해 산정
- 기능 중심으로 프로젝트 산정하는 방법

## ❖ 장점

- 개발 환경(개발 언어, 도구 등)에 독립적

## ❖ 단점

- 계산이 주관적인 자료를 바탕으로 한다.
- 계산하기에 어렵고 자동화하기에 힘들다

# FP(Function Point) (2/5)

---

## ❖ 산정 프로세스

- 1. 기능 식별
- 2. 조정 전 기능점수(UFP: Unadjusted Function Point) 계산
- 3. 조정 인자 결정
- 4. 기능점수 공식에 대입하여 조정인자가 반영된 조정 후 기능점수 계산

# FP(Function Point) (3/5)

---

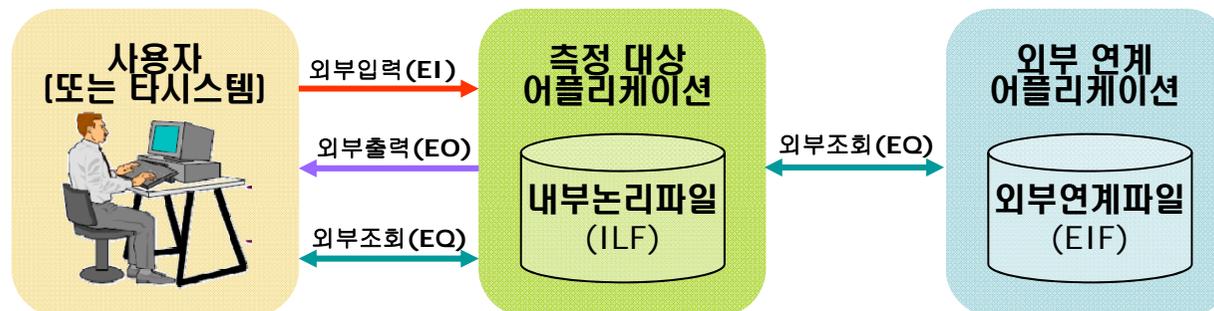
## ❖ 1. 기능 식별

- 5가지 기능 유형으로 기능을 분류하고 복잡도 매트릭을 통해 각 기능의 복잡도 (단순, 평균, 복잡) 결정
- 복잡도는 어떻게 선택하나?
  - 각 기능 유형별로 복잡도를 선택할 수 있는 매트릭을 이용
  - 기능점수 및 매트릭 정보: 한국정보기술원가표준원(<http://www.kfpug.or.kr/>) 참조

# FP(Function Point) (3/5)

## ❖ 5가지 기능 유형

- **내부 논리 파일: ILF (Internal Logical Files)**
  - 측정 어플리케이션 내부에서 유지되면서, 사용자가 식별 가능한 데이터 그룹 또는 제어정보의 개수를 계산.
  - 사용자가 해당 파일의 내용을 조회, 수정, 갱신, 삭제.
- **외부 연계 파일: EIF (External interface Files)**
  - 다른 어플리케이션의 내부에서 유지, 사용자가 식별 가능한 데이터 그룹 또는 제어정보의 개수를 계산.
  - 외부연계파일은 수정, 삭제 등의 활동은 할 수 없고 내용만 참조.
- **외부 입력: EI (External Input)**
  - 어플리케이션 경계 밖에서 들어오는 데이터 및 제어정보가 내부논리파일을 유지보수, 시스템 행위를 변경하는 기능의 개수를 계산.
  - 새로운 내용을 등록하는 기능 등이 해당.
- **외부 출력: EO (External Output)**
  - 내부논리파일과 외부연계파일로부터 데이터나 제어 정보를 가져와 처리 로직(연산 등)을 거친 후 어플리케이션 경계 밖으로 전송하는 기능의 개수를 계산.
  - 보고서, 오류 메시지 등이 해당.
- **외부 조회: EQ (External Queries)**
  - 내부논리파일과 외부연계파일로부터 데이터나 제어정보를 그대로 어플리케이션 경계 밖으로 전송하는 기능의 개수를 계산.
  - 검색, 조회 기능.



# FP(Function Point) (4/5)

## ❖ 2. 조정 전 기능점수(UFP: Unadjusted Function Point) 계산

- 아래의 표와 같이 매트릭을 통해 얻어진 각 기능 유형별 기능 점수를 합한 점수
- 소프트웨어 개발 특성을 포함하지 않은 점수

기능타입 복잡성	단순	평균	복잡
내부논리파일	○ × 7	○ × 10	○ × 15
외부연계파일	○ × 5	○ × 7	○ × 10
외부입력	○ × 3	○ × 4	○ × 6
외부출력	○ × 4	○ × 5	○ × 7
외부조회	○ × 3	○ × 4	○ × 6

# FP(Function Point) (5/5)

## ❖ 3. 조정 인자 결정

- 14가지 시스템 특성을 고려하여 조정하며 각 항목이 시스템 개발에 미치는 영향의 정도를 0~5까지 평가하여 그 합을 구함

구분	0	1	2	3	4	5
데이터 통신						
분산 데이터 처리						
시스템 성능						
자원 제약 제도						
트랜잭션 비율						
온라인 데이터 입력						
최종 사용자 효율성						
온라인 갱신						
처리 복잡도						
재사용성						
설치 용이성						
운영 용이성						
다중 설치성						
변경 용이성						

Fi 값 테이블

# FP(Function Point) (5/5)

---

## ❖ 4. 기능점수 공식에 대입하여 조정인자가 반영된 조정 후 기능 점수 계산

### - 공식

$$FP = UFP \text{ total} \times [ 0.65 + 0.01 \times \sum(F_i) ]$$

# 일정 계획

---

# 일정 계획

---

## ❖ 의미

- WBS를 이용하여 구성한 하위 작업들을 산정을 통해 나온 기간이나 비용에 맞도록 계획하는 활동

## ❖ 표현 방법

- 차트를 사용
- 장점
  - 차트 자체의 비주얼적 특징 및 역할 할당
  - 병렬 작업 구성 등 일정 구성에 도움을 받음
- 종류
  - 퍼트(PERT) 차트
  - 간트(Gantt) 차트

# 퍼트 차트(PERT Chart)

---

## ❖ 소개

- PERT: Program Evaluation and Review Technique
- 프로젝트를 구성하는 작업들 사이의 관계 및 흐름을 그래픽으로 표현

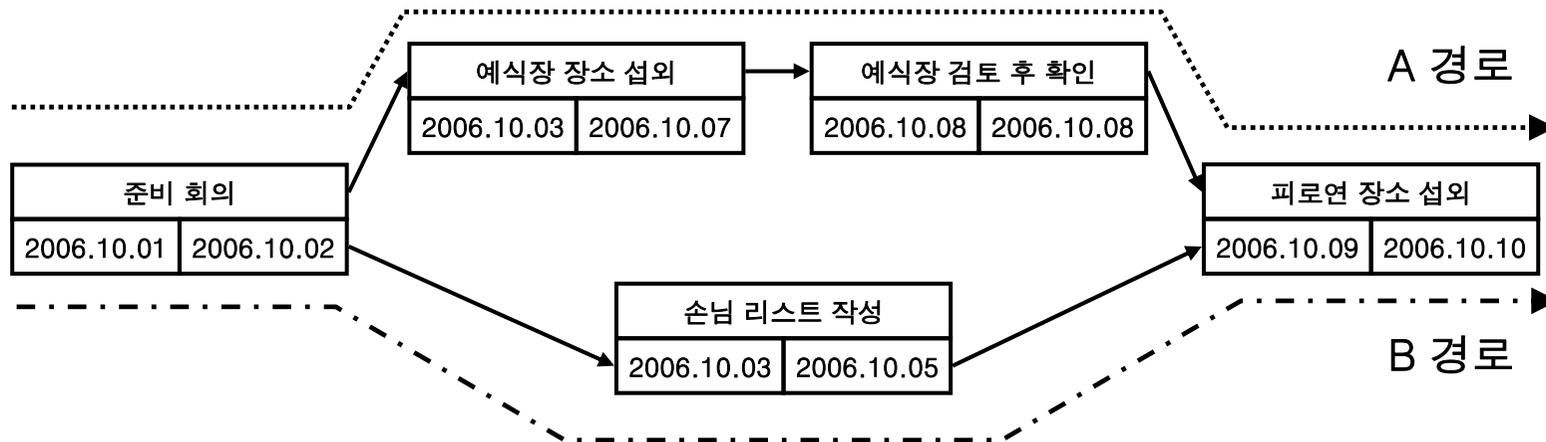
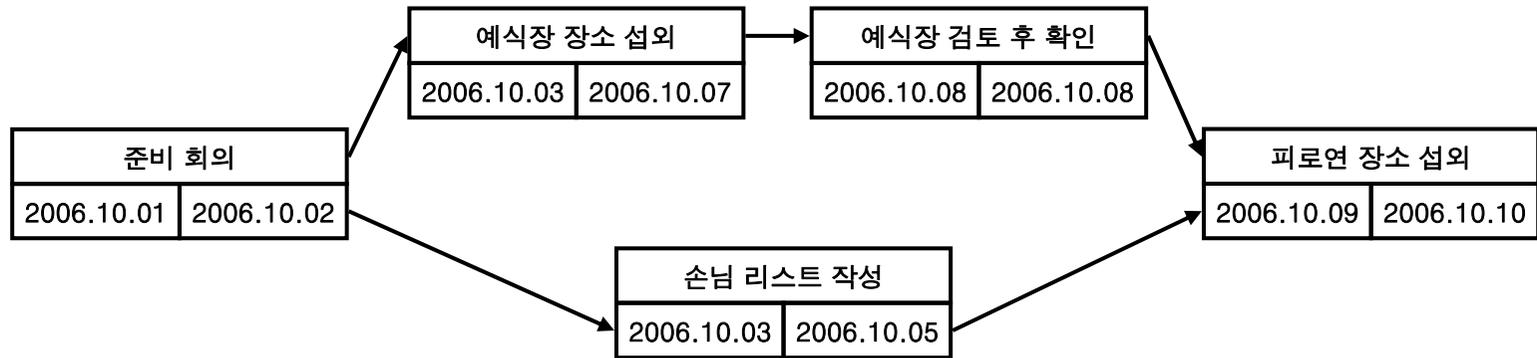
## ❖ 표기법과 의미

- 박스(또는 원): 작업이나 업무
- 선과 화살표: 각 작업간의 순서와 의존성을 표현
- 각 작업은 병행적으로 수행될 수 있음

## ❖ 장점

- 관리자는 프로젝트의 모든 작업들 간의 상호 의존성 및 프로젝트가 진행되는 다양한 경로 파악 가능
- 관리자는 프로젝트가 종료되는데 가장 적은 시간으로 퍼트 차트의 가장 긴 경로 예측 가능

# 퍼트 차트의 예제 – 결혼식 준비



# 간트 차트(Gantt Chart)

---

## ❖ 소개

- Henry L. Gantt에 의해 제안됨
- 프로젝트의 일정, 예산 및 자원 계획 등을 목적으로 사용될 수 있는 프로젝트 제어 기법

## ❖ 표기법과 의미

- 시간의 순으로 되어 있는 캘린더에 프로젝트 시작 시간과 종료 시간을 막대 형태로 나타냄
  - 왼쪽 열: 수행해야 할 작업들, 시작일과 완료일, 기간 표시
  - 막대의 길이: 수행해야 하는 작업의 시간

## ❖ 장/단점

- 퍼트 차트와 달리 작업간의 의존성을 보여주지는 않음
- 차트를 왼쪽에서 오른쪽으로 읽으면 작업 시작일과 종료일을 분명히 알 수 있음
  - 현재 작업 상태, 늦어진 작업 현황, 앞으로 진행할 작업에 대해 쉽게 파악 가능

# 간트 차트의 예 – 결혼식 준비

ID	작업 이름	시작	완료	기간	2006년 10월										
					1	2	3	4	5	6	7	8	9	10	11
1	준비회의	2006-10-01	2006-10-02	2d	[Blue bar from Day 1 to Day 2]										
2	예식장 장소섭외	2006-10-03	2006-10-07	5d	[Blue bar from Day 3 to Day 7]										
3	예식장 검토후 확인	2006-10-08	2006-10-08	1d	[Blue bar on Day 8]										
4	손님 리스트 작성	2006-10-03	2006-10-05	3d	[Blue bar from Day 3 to Day 5]										
5	피로연 장소 섭외	2006-10-09	2006-10-10	2d	[Blue bar from Day 9 to Day 10]										

# 위험 관리(Risk Management)

---

# 위험 관리란?

---

## ❖ 의미

- 프로젝트의 위험 요소들을 인식하고 그 영향을 분석, 관리하는 활동

## ❖ 위험의 의미

- 프로젝트의 결과가 잘못되는 데에 심각한 영향을 주는 요소
- 예) 프로젝트 기간 중 개발 인력의 부족, 갑작스런 요구사항의 변경, 예정되었던 기술의 실패 등

# 위험 관리

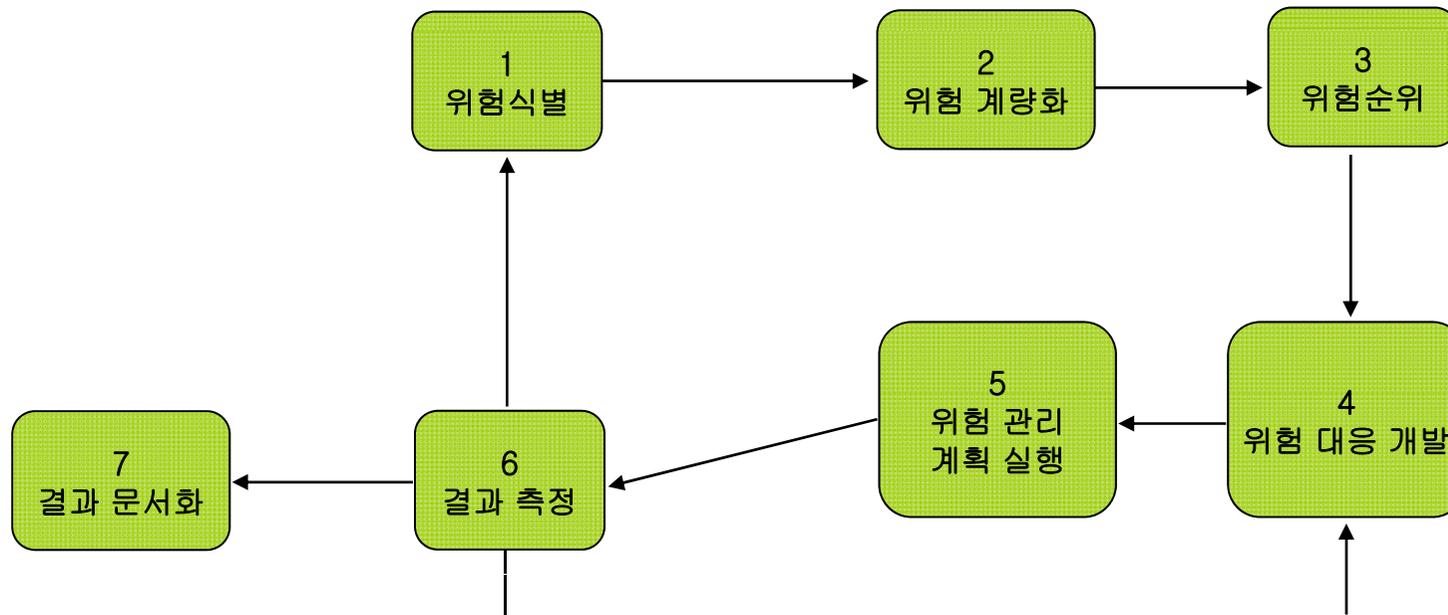
---

## ❖ 중요성

- 위험 요소에 대해 적절한 관리를 통해 프로젝트 실패를 예방
- 초기 단계에 위험을 분석하였다더라도 위험관리 계획이 있어야 실패를 예방

# 위험 관리 과정

## ❖ 위험 관리 과정



# 위험 요소 분석

---

## ❖ 위험 요소 분석의 예

위험 요소	발생 확률	위험 영향	관리 수위
요구사항 변경	90%	대	A
기술 실패	40%	대	B
인력 이탈	60%	중	B

# 연습문제

---

1. 프로젝트 계획 수립을 시작할 때 제일 먼저 해야 하는 작업을 기술하라.
2. 프로젝트의 개발 비용 산정 시 결정에 영향을 주는 요소를 3가지 기술하라.
3. 소프트웨어 개발 비용은 시스템 크기, 개발 기간, 신뢰도, 투입 인력들과 일정한 상관 관계가 있다. 개발 비용을 Y 축으로 하는 반비례 그래프를 그려본다면, X 축으로 어떤 요소가 가장 타당한지 기술하라.
4. 두 명의 개발자가 5개월에 걸쳐 10,000라인의 코드를 개발하였을 때, 월별(Man Month) 생산성 측정을 위한 계산 방식은 어떻게 하는지, 식을 기술하라.
5. LOC 기법에 의하여 예측된 총 라인수가 25,000라인일 경우 개발에 투입될 프로그래머의 수가 5 명이고, 프로그래머들의 평균 생산성이 월당 500라인일 때, 개발에 소요되는 시간을 계산하라.
6. 대표적인 소프트웨어 산정 모형(Estimation Model) 3가지를 기술하라.

# 팀 프로젝트

---

6, 7, 9주차

# 이번 주 할일

---

- ❖ 각 팀은 요구사항 명세서를 작성하여 제출한다.
- ❖ 각 팀은 프로젝트 계획서를 작성하여 제출한다.

# 다음 주 제출 문서

---

- ❖ 설계 문서를 작성하여 제출한다.