



소프트웨어 프로젝트 관리

가을, 2024

제홍@충북.ac.kr

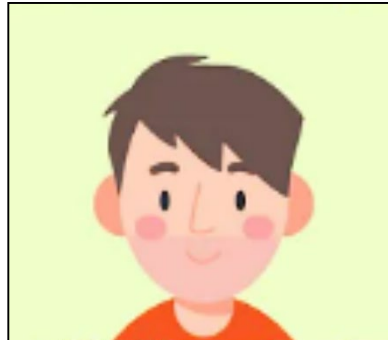
프로로그

당신은 어떤 유형의 리더를 선호하시나요?



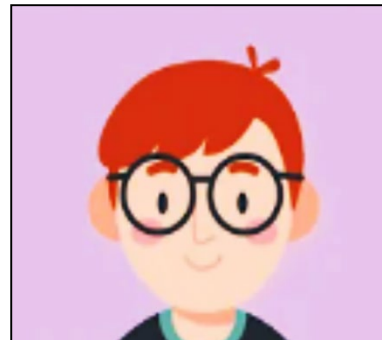
멍부

멍청함 + 근면함



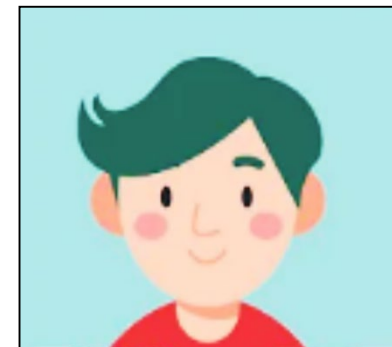
멍게

멍청하고 게으르다



똑부

똑똑하고 근면하다



똑게

똑똑하고 게으르다

(?) 똑게 >> 멍게 >> 똑부 >> 멍부

다루는 주제

소프트웨어 프로젝트 관리

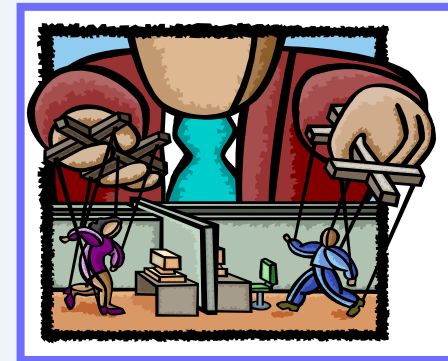
- 프로젝트 관리란 무엇인가요?
- 프로젝트 계획
- 프로젝트 비용 추정
 - 기능 포인트
 - 코코모

프로젝트 제어 기술

- 작업 분할 구조
- 간트 차트
- PERT 차트

프로젝트 팀 조직 위험 관리

프로젝트 관리 계획



소개..

소프트웨어 프로젝트는 왜 실패하는가?

Lack of SW mind



Lack of appropriate SE skills



Insufficient software project management



소프트웨어 프로젝트에서는 어떤 문제를 관리해야 합니까?

프로젝트 관리

필요

- 비효율적인 관리로 인한 프로젝트 실패
- 프로젝트 지연, 신뢰할 수 없음, 예산 초과, 성과 저하

다른 유형의 엔지니어링과의 차이점

- 제품은 무형입니다
- SW 프로세스에 대한 명확한 이해가 없음
- 제조 집약형과 대조적으로 설계 집약형

성공적인 소프트웨어 프로젝트에 가장 중요한 기여자

".. 우리가 사용하는 도구가 아니라 사람입니다."

"... 똑똑한 사람들이 있다는 것.. 제 생각엔 다른 건 별로 중요하지 않아요."

"경영에 있어서 내가 지키는 유일한 원칙은 좋은 사람들을 확보하는 것입니다."

관리 기능

경영의 정의

- 생성 및 유지 관리** 개인들이 그룹으로 함께 일하면서 그룹 목표 달성을 위해 효율적이고 효과적으로 수행할 수 있는 기업 내부 환경

경영의 일반 기능

- 계획: 목표, 자원, 정보 흐름, 사람, 유물
- 조직화 : 그룹에 대한 권한과 책임
- 인력 채용: 인력 채용
- 지휘 : 부하직원을 지휘하다
- 통제: 활동 측정 및 수정

관리 단계

1단계. 계획

- 목표를 이해하고 문서화합니다.
- 일정, 예산 및 기타 리소스 요구 사항 개발

2단계. 자원 획득

- 우주, 컴퓨팅 자원, 재료 및 인적 자원

3단계. 실행

- 계획을 실행에 옮기다.

4단계. 모니터링

- 프로젝트 진행 상황을 확인합니다.
- 계획에서 벗어난 사항을 처리하기 위해 필요한 조치를 취합니다.

프로젝트 계획

계획 문제

- 가정, 목표 및 제약 조건을 명확하게 정의하고 문서화합니다.
- 필요한 리소스와 예산을 결정하세요
 - 사람들의 수와 기술 수준
 - 컴퓨터 리소스의 양

계획에서의 예측

- 엔지니어가 몇 명이나 필요할까요?
 - 생산력 소프트웨어 엔지니어의
- 소프트웨어 비용 추정

소프트웨어 생산성

프로젝트 계획을 위해서는 다음이 필요합니다.

- 작업의 어려움을 추정합니다.
- 각 엔지니어가 얼마나 많은 작업을 해결할 수 있는지 추정해 보세요.

생산성 지표

- 기능성의 양
- LOC: 이상적인 생산성 측정 기준이 아님

기능의 개념을 정량화하는 방법은 무엇인가?

- 기능 점수.

코드 크기(LOC)

생산성을 측정하는 데 가장 일반적으로 사용되는 지표. 가장 일반적인 두 가지 코드 크기:

- DSI(전달된 소스 지침)
 - 고객에게 전달되는 코드 줄만.
- NCSS(주석이 없는 소스 진술)
 - 주석 줄은 계산되지 않습니다.

해결해야 할 문제가 많지만 측정하기 쉽습니다. 현재 유용하지는 않지만 참조합니다.

기능 포인트

소프트웨어 시스템의 기능을 정량화하려고 시도합니다. 시스템의 복잡성을 특성화합니다.

예측에 사용할 수 있습니다

- 얼마나 걸릴까

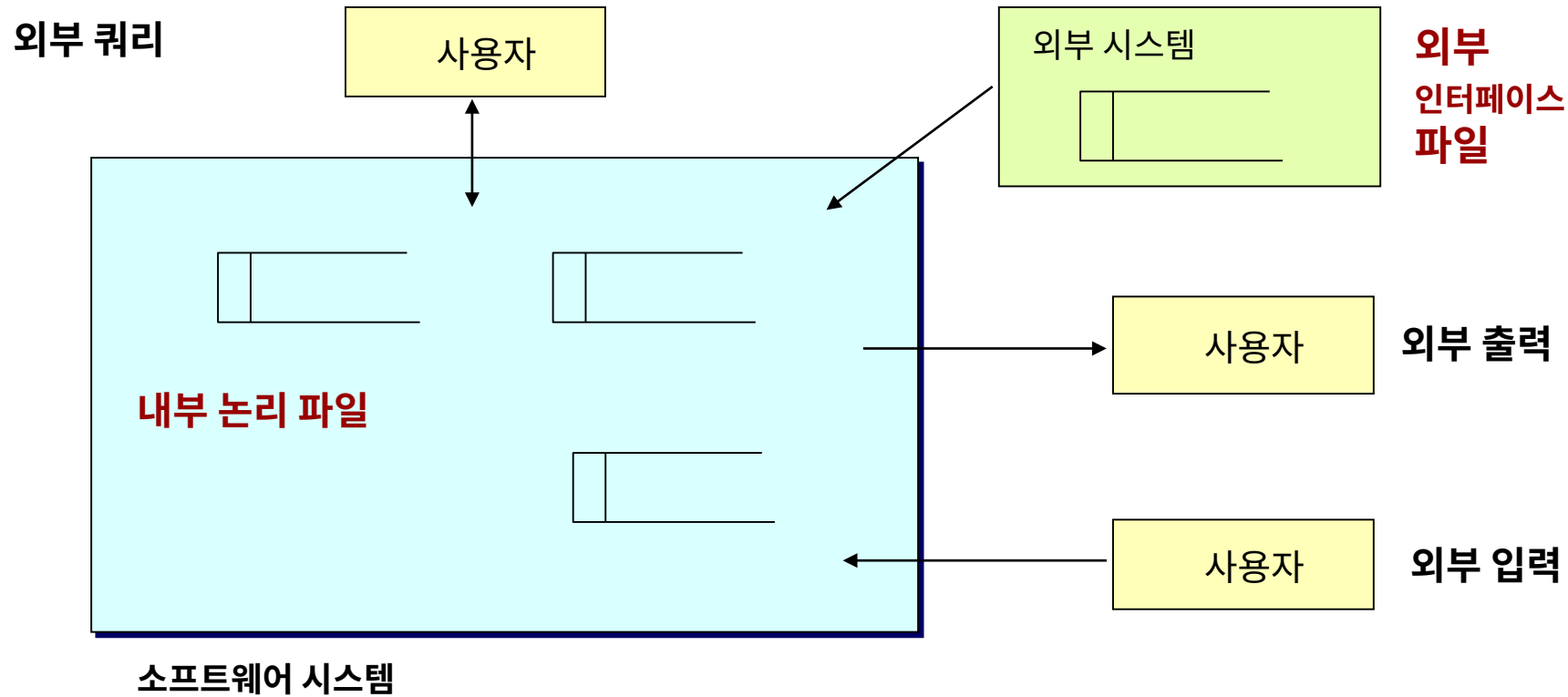
- 그것을 하려면 얼마나 많은 사람이 필요한가.

정보처리 애플리케이션에 적합합니다.

생산성, 금액, 오류 수를 측정할 수 있습니다. 미래 계획의 기초로 사용할 수 있습니다.

다양한 언어의 상대적 힘을 측정하는 데 사용됩니다. 해결해야 할 문제가 많고, 전망이 좋지 않습니다.

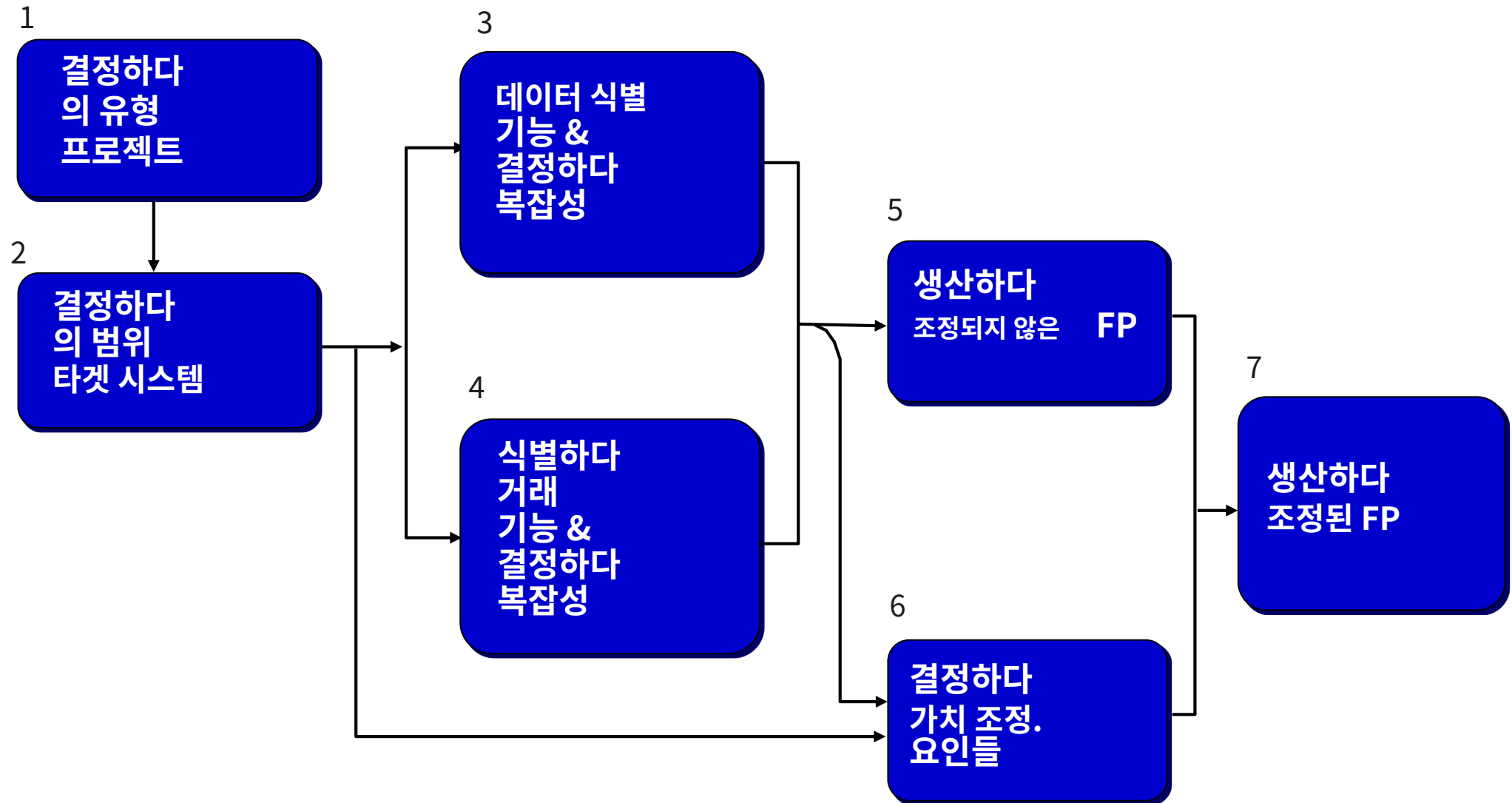
일반 소프트웨어 시스템의 기능



-데이터 기능 : 내부 논리 파일 + 외부 인터페이스 파일

-거래 기능 : 외부 쿼리 + 외부 출력 + 외부 입력

FP 분석 방법



각 단계의 활동

1. 프로젝트 유형 결정

-신규 프로젝트 / 유지 관리 프로젝트 / 개선 프로젝트

2. 타겟 시스템의 범위 결정

-전체 또는 일부 / 자체 개발, 아웃소싱, 패키지 인수, ...

3. 데이터 기능 및 복잡성 식별

내부 논리 파일		데이터 요소 유형		
		1 - 19	20 - 50	> = 51
기록 요소 유형	1 2 - 5 > 5	낮은 낮은 중간	낮은 중간 높은	중간 높은 높은

각 단계의 활동(계속)

4. 거래 기능과 그 복잡성을 식별합니다.

-예: "외부 입력"에 대한 복잡도 행렬

외부 입력		데이터 요소 유형		
		1 - 4	5 - 15	> = 16
참조	< 2	낮은	낮은	중간
파일	2	낮은	중간	높은
유형	> 2	중간	높은	높은

5. 조정되지 않은 FP 생성

-UFP = 가중치의 합
모든 기능에 대해

기능	복잡성 수준(가중치)		
	낮은	중간	높은
국제전기통신연합(ILF)	엑스 7	엑스 10	엑스 15
이프	엑스 5	엑스 7	엑스 10
아이	엑스 3	엑스 4	엑스 6
이오	엑스 4	엑스 5	엑스 7
이큐	엑스 3	엑스 4	엑스 6

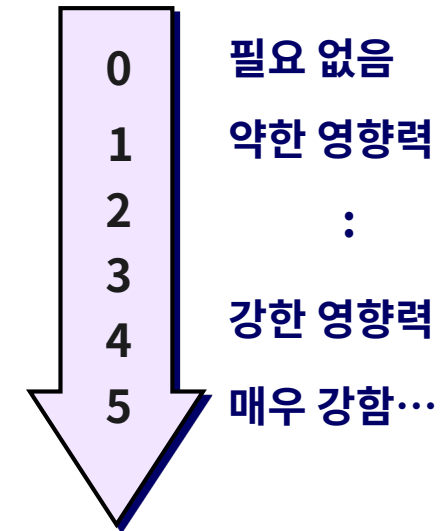
각 단계의 활동(계속)

6. 가치 조정 계수(VAF) 결정

-14 시스템 특성

1. 데이터 통신	8. 온라인 업데이트
2. 분산 데이터 처리	9. 복잡한 처리
3. 성능	10. 재사용성
4. 많이 사용되는 구성	11. 설치 용이성
5. 거래율	12. 조작의 용이성
6. 온라인 데이터 입력	13. 여러 사이트
7. 최종 사용자 효율성	14. 변화를 촉진하다

• 영향력의 정도



$$VAF = (TDI * 0.01) + 0.65$$

7. 조정된 FP를 생성합니다.

-조정된 기능 포인트, AFP = UAF엑스VAF

프로그래밍 언어별 LOC 대 FP

구현 코드 줄 / 1 FP

구현 언어		# 라인	구현 언어	# 라인
집회	기초적인	320	포트란	107
	매크로	213		
기초적인		107	HTML3.0	15
비주얼 베이직		29	리스프	64
기름		132	자바	53
씨++		53	폴란드/나	80
코볼		107	SQL	13
델파이		29	파워 빌더	16

* 출처 : www.theadvisors.com/langcomparison.htm

생산성에 영향을 미치는 다른 요소

인력의 역량 제품의 복잡성 요구되는 신뢰성

-시간 제약(실시간 시스템)

일정 제약

언어 경험

인사이동

시스템의 재구조화...

소프트웨어 비용 추정 기술

알고리즘 비용 모델링

- 코코모, 코코모 2

전문가의 판단

유추에 의한 추정

- 비슷한 프로젝트로

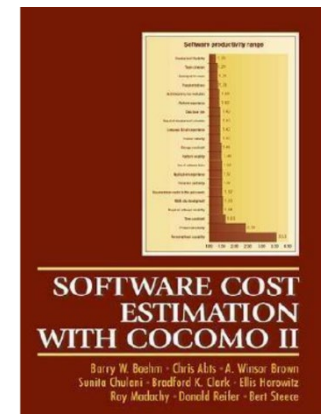
파킨슨의 법칙

- 사용 가능한 리소스에 따라 결정됨

탐다운 추정

- 전반적인 기능, 첫 번째

하향식 추정



코코모

구성적 비용 모델, 1981 B. Boehm 제공된 소스 지침
을 기반으로 KDSI

점점 더 복잡해지는 세 가지 다른 모델 세트를 기반으로 함
그리고 세부 사항의 수준

명목 노력 및 일정 방정식

개발 모드	명목상의 노력	일정
본질적인	$PM = 2.4(KDSI)^{1.05}$	$TDEV = 2.5(\text{오후})_{\text{개발자}}^{0.38}$
반쯤 떨어진	$PM = 3.0(KDSI)^{1.12}$	$TDEV = 2.5(\text{오후})_{\text{개발자}}^{0.35}$
내장된	$PM = 3.6(KDSI)^{1.20}$	$TDEV = 2.5(\text{오후})_{\text{개발자}}^{0.32}$

PM = 사람(프로그래머)-월, $PM_{\text{개발자}} = PM * \text{노력 배수}$,

$TDEV$ = 프로젝트를 완료하는 데 필요한 개월 수

$\text{오후}_{\text{개발자}} / TDEV = \text{필요한 인원 수}$

코코모

노력 배수기

$$= \prod_{L=1}^{15} C/L$$

Cost Driver(C/L)	Ratings					
	Very low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	.75	.88	1.00	1.15	1.40	
Data base size		.94	1.00	1.08	1.16	
Product complexity	.70	.85	1.00	1.15	1.30	1.65
Computer attributes						
Execution time constraints			1.00	1.11	1.30	1.66
Main storage constraints						
Platform volatility		.87	1.00	1.15	1.30	
Computer turn around time		.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	.86	.71	
Applications experience	1.29	1.13	1.00	.91	.82	
Programmer capability	1.42	1.17	1.00	.86	.70	
Virtual machine experience*	1.21	1.10	1.00	.90		
Programming language experience	1.14	1.07	1.00	.95		
Project attributes						
Use of modern programming practices	1.24	1.10	1.00	.91	.82	
Use of software tools	1.24	1.10	1.00	.91	.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

코코모(계속)

민감도 분석이 가능합니다

- 노력 배수의 매개변수를 변경하여

COCOMO II와의 차이점

- 프로세스 모델에 대한 가정
 - COCOMO : 순차적 개발 프로세스(폭포수 모델)
 - COCOMO II : 반복적 접근 방식, RAD, 재사용 중심 접근 방식 등
- 평가자
 - COCOMO : 소스 지침(KDSI)
 - COCOMO II : 소스 지침 및 기능 포인트

코코모 2

B. 보엠, 1995

프로젝트 진행 상황에 따라 다른 모델 사용

-노력 배수의 매개변수를 변경하여

COCOMO II의 3가지 모델

	1단계	2단계	3단계
언제	프로토타입 단계	예비 설계	건축 이후
메트릭	앱 포인트	FP	FP 및 LOC
재사용	절대적인	명백한	명백한
요구 사항 변경	절대적인	비용 요인 중 하나	비용 요인 중 하나

* 적용 포인트 : 입출력 인터페이스의 구성요소 수, 화면 수

프로젝트 제어

활동 진행 상황을 모니터링합니다.

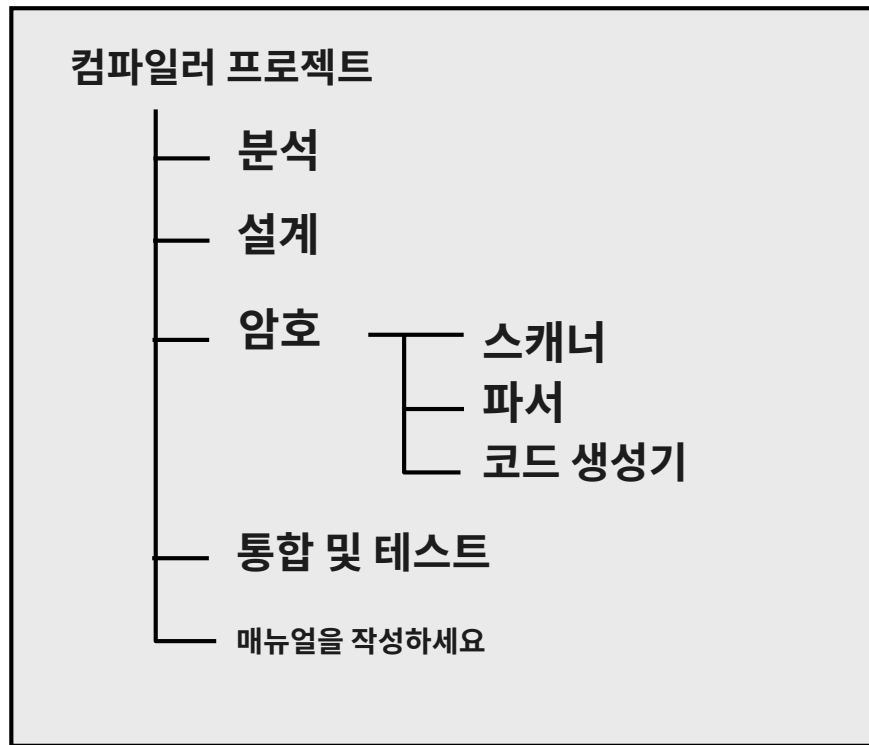
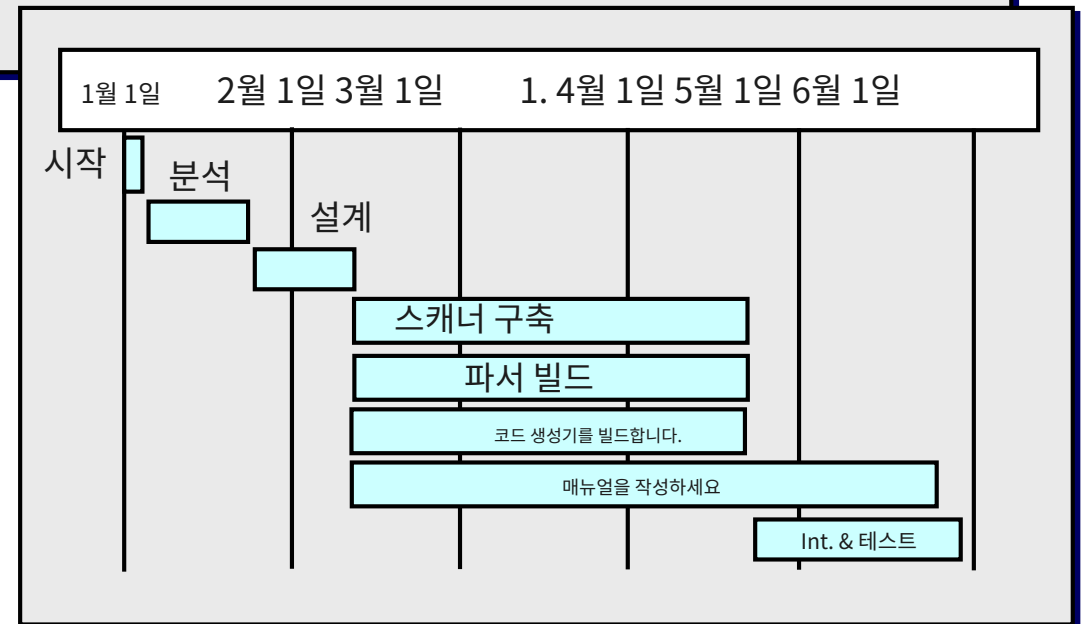
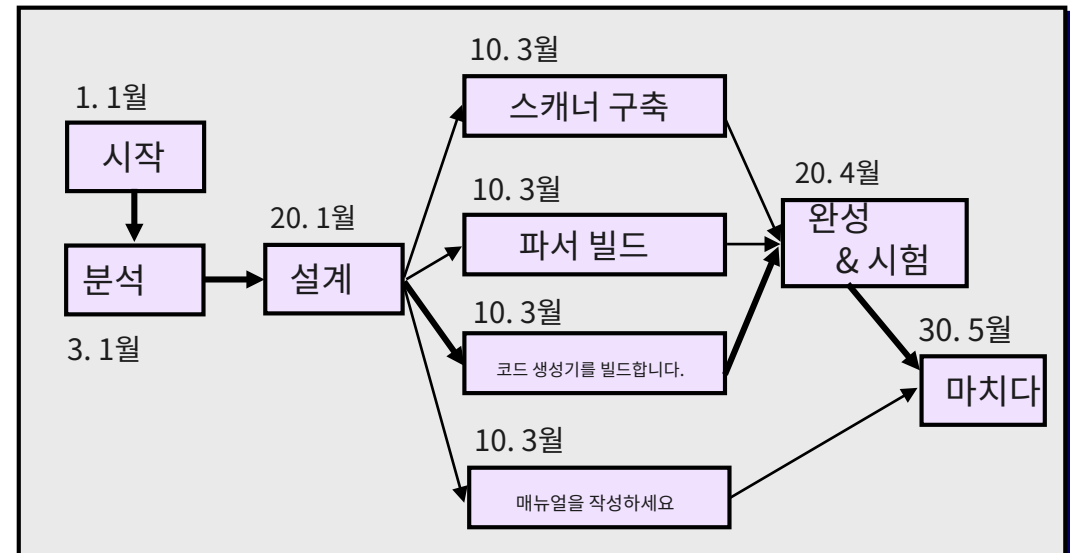
계획에서 벗어나는 일이 발생하는 시점을 감지합니다. 프로젝트 제어 기술을 사용합니다.

- 작업 분할 구조
- 간트 차트
- PERT 차트(프로그램 평가 및 검토 기법)
- 칸반 보드

프로젝트 제어 기술

WBS(작업분류체계) 간트차트

PERT 차트












프로젝트 제어 기술 - WBS (1)

WBS(작업분류체계)

- 프로젝트 목표를 여러 개의 중간 목표로 세분화하여
- WBS 목표
 - 프로젝트에서 수행해야 하는 모든 활동을 식별합니다.
- 프로젝트의 주요 활동으로 루트가 표시된 트리
 - 더 작은 구성요소로 분해됨
 - 나무의 각 잎이 관리자가 크기, 난이도 및 리소스 요구 사항 측면에서 추정할 수 있다고 확신하는 작업을 나타낼 때까지
- 에 사용됨
 - 프로젝트 계획 요약
 - 스케줄링 프로세스에 대한 입력

프로젝트 제어 기술 - WBS (2)

WBS의 예

ID		작업 %	작업 이름	기간	시작 날짜	완료 날짜	선행 작업	자원 이름
1		0%	오온행 SPI Project	424 일	04-01-05 (월)	05-08-29 (월)		
2		0%	현황진단단계	17 일	04-01-05 (월)	04-01-30 (금)		
3		0%	SPI TFT 구성	1 일	04-01-05 (월)	04-01-05 (월)		
4		0%	SPI 운영위원회 구성	3 일	04-01-06 (화)	04-01-08 (목)	3	
5		0%	SPI 세미나	2 일	04-01-06 (화)	04-01-07 (수)	3	
6		0%	현황진단	9 일	04-01-08 (목)	04-01-20 (화)	3,7	
7		0%	프로세스 교육	2 일	04-01-06 (화)	04-01-07 (수)	3	
8		0%	PIT 구성	4 일	04-01-09 (금)	04-01-14 (수)	4	
9		0%	우선순위 결정	1 일	04-01-26 (월)	04-01-26 (월)	6	
10		0%	SPI 계획 수립 및 승인	5 일	04-01-26 (월)	04-01-30 (금)	6	
11		0%	SPI TFT 멘터링	15 일	04-01-06 (화)	04-01-29 (목)	3	
12		0%	단계 종료 보고	1 일	04-01-30 (금)	04-01-30 (금)	11	
13		0%	구축 및 초기적용 단계	126 일	04-02-02 (월)	04-07-30 (금)	2	
14		0%	표준 프로세스 개발	70 일	04-02-02 (월)	04-05-12 (수)		
15		0%	프로세스 개발	40 일	04-02-02 (월)	04-03-29 (월)		
16		0%	템플릿 개발	12 일	04-03-30 (화)	04-04-15 (목)	15	
17		0%	가이드라인 개발	12 일	04-04-16 (금)	04-05-03 (월)	16	
18		0%	체크리스트 개발	6 일	04-05-04 (화)	04-05-12 (수)	17	
19		0%	방법론 개발	50 일	04-02-02 (월)	04-04-13 (화)	2	
20		0%	객체지향 방법론	25 일	04-02-02 (월)	04-03-08 (월)		
21		0%	정보공학 방법론	25 일	04-03-09 (화)	04-04-13 (화)	20	
22		0%	SPI 교육	12 일	04-05-13 (목)	04-05-31 (월)	14,19	
23		0%	방법론 교육	4 일	04-05-13 (목)	04-05-18 (화)	14,19	
24		0%	프로세스 파일럿 적용	43 일	04-06-01 (화)	04-07-29 (목)	22,23	
25		0%	프로젝트 멘터링 및 코칭	43 일	04-06-01 (화)	04-07-29 (목)	22,23	
26		0%	단계 종료 보고	1 일	04-07-30 (금)	04-07-30 (금)	24,25	

프로젝트 제어 기술 - 간트 차트(1)

일정, 예산 책정에 사용할 수 있는 기술
자원 계획.

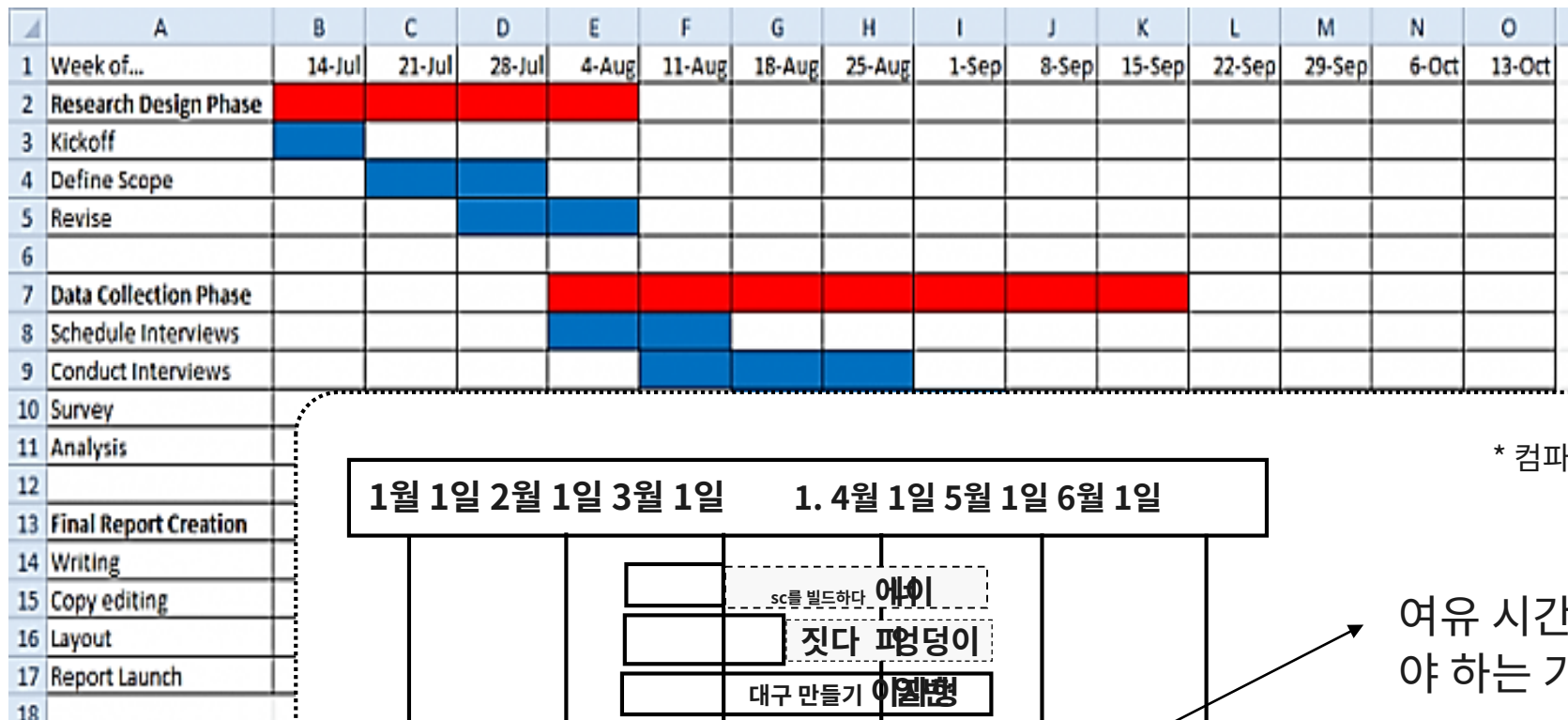
일종의 막대형 차트

- 각 막대는 활동을 나타냅니다
- 활동에 계획된 시간 길이에 비례하여 타임라인에 따라 그려짐

자원 할당 및 인력 계획에 사용됨 작업 간 종속성
을 강조하지 마십시오.

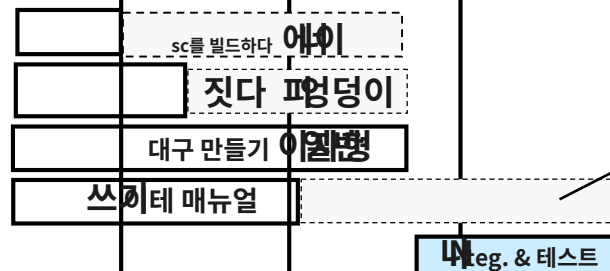
프로젝트 제어 기술 - 간트 차트(2)

간트 차트의 예



* 컴파일러 프로젝트 *

1월 1일 2월 1일 3월 1일 1. 4월 1일 5월 1일 6월 1일



여유 시간 : 작업을 완료해야 하는 가장 늦은 시간

프로젝트 관리 기술 - PERT 차트(1)

프로그램 평가 및 검토 기술, 건방진
크리티컬 패스 방법, CPM

수행해야 할 활동의 종속성을 나타냅니다.
프로젝트

- 상자(활동)와 화살표(활동의 종속성)의 네트워크

눈에 띄는 특징

- 활동에서 가능한 모든 병렬성을 노출합니다.
- 대체 일정의 스케줄링 및 시뮬레이션을 허용합니다.
- 관리자가 프로젝트를 모니터링하고 제어할 수 있도록 합니다.

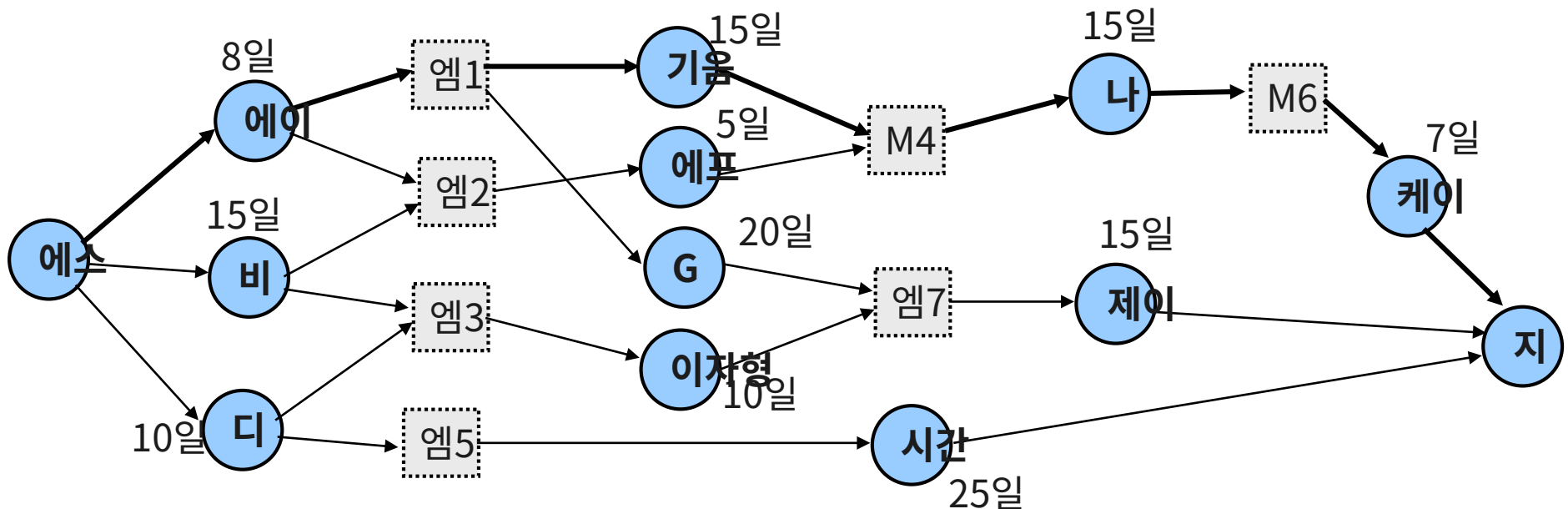
프로젝트 관리 기술 - PERT 차트(2)

PERT 차트의 날짜

- 가장 빠른 시작 날짜와 가장 늦은 시작 날짜
- 가장 빠른 완료 날짜와 가장 늦은 완료 날짜

중요 경로

- 어떠한 활동의 지연으로 인해 전체 프로젝트가 지연되는 경로



칸반 보드

칸반 보드

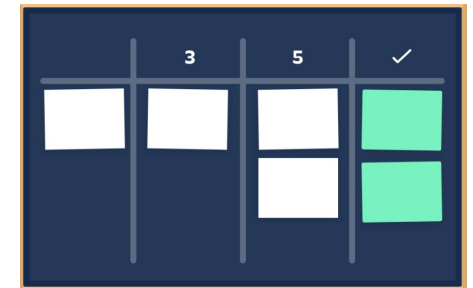
- 작업 시각화, 진행 중인 작업 제한, 효율성 극대화에 도움이 되도록 설계된 관리 도구
- 병목 현상과 과도한 약속을 식별하는 데 도움이 되는 진행 중인 작업을 보여주는 시각화 도구

칸반의 기원

- JIT 생산을 위한 Toyota Kanban 시스템(1963)
 - 물류, 공급업체 관리, 고객 배송 등 제조 운영을 구성하는 데 사용됩니다.

적용 분야

- Agile 및 DevOps 팀이 일상 업무에 질서를 확립합니다.
- 마케팅팀, HR팀, 개인 칸반, ..



칸반 보드

JIRA의 디지털 칸반 보드

The screenshot displays the Jira Kanban Board for the 'Beyond Gravity' project. The board is organized into four columns: TO DO (6 items), IN PROGRESS (6 items), IN REVIEW (6 items), and DONE (6 items). Each item is a card showing the task name, category (e.g., BILLING, ACCOUNTS, FORMS, FEEDBACK), ID (e.g., NUC-344, NUC-342), and assignee. The left sidebar shows project navigation options like Roadmap, Backlog, Board, Reports, Issues, Code, Security, Releases, and Project settings. The top navigation bar includes Jira logo, 'Your work', 'Projects', 'Filter', 'Dashboards', 'Teams', 'Plans', 'Apps', and a 'Create' button. A search bar and user avatars are also visible.

Column	Item	Category	ID	Assignee
TO DO (6)	Optimize experience for mobile web	BILLING	NUC-344	[Avatar]
	Onboard workout options (OWO)	ACCOUNTS	NUC-360	[Avatar]
	Multi-dest search UI mobileweb	ACCOUNTS	NUC-337	[Avatar]
	Billing system integration - frontend	FORMS	NUC-339	[Avatar]
	Account settings defaults	ACCOUNTS	NUC-340	[Avatar]
IN PROGRESS (6)	Fast trip search	ACCOUNTS	NUC-342	[Avatar]
	Affelite links integration - frontend	BILLING	NUC-335	[Avatar]
	Shopping cart purchasing error - quick fix required	FORMS	NUC-341	[Avatar]
IN REVIEW (6)	Revise and streamline booking flow	ACCOUNTS	NUC-367	[Avatar]
	Travel suggestion experiments	ACCOUNTS	NUC-358	[Avatar]
	Ongoing customer satisfaction	ACCOUNTS	NUC-354	[Avatar]
	Planet Taxi Device exploration & research	FEEDBACK	NUC-351	[Avatar]
DONE (6)	High outage: Software bug fix - BG Web-store app crashing	BILLING	NUC-340	[Avatar]
	Web-store purchasing performance issue fix....t	FORMS	NUC-341	[Avatar]

계획에서의 편차 처리

관리자는 편차를 어떻게 처리할지 결정해야 합니다.
일정

- 엔지니어를 추가하지 않음, 하지만 엔지니어는 맞음
- 임시로 수석 엔지니어를 재배치하거나 전문 문제 해결 컨설턴트를 고용합니다.
- 요구 사항 스크러빙
- 원래 계획과 일정이 잘못되었다는 것을 인정합니다.
- 지연을 올바른 행동 방향으로 받아들이십시오

팀 조직

목적 : 공통 목표를 향한 협력을 촉진하기 위한 조직 구조 유형

- 중앙 제어 팀
- 분산 제어 팀
- 그리고 더 많은 것..

cf) 기능 중심 구조, 프로젝트 중심 구조 및 매트릭스 구조

조직 구조 선택에 영향을 미치는 고려 사항

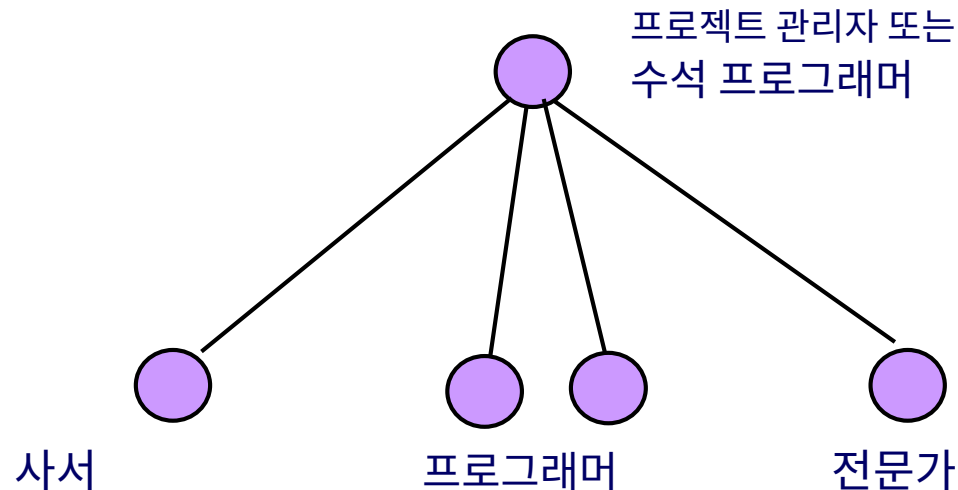
- 프로젝트의 길이
- 업무의 특성과 커뮤니케이션 양
- 팀에 적합한 크기
 - 팀은 충분히 커야 하지만 너무 커서는 안 되고, 충분히 작아야 하지만 너무 작아서도 안 됩니다.

조직 구조

중앙 제어 팀

- 최고 프로그래머 팀
- 수석 프로그래머는 프로젝트의 설계 및 모든 기술적 세부 사항에 대한 책임을 집니다.
- 작업을 잘 이해하면 효과적입니다.
- 수석 프로그래머는 업무가 너무 과중할 수 있습니다.

의사소통 패턴:

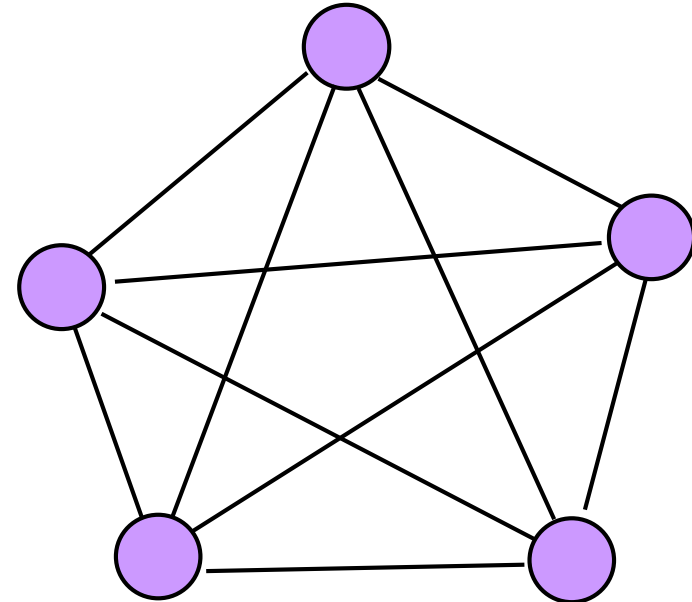


조직 구조 (계속)

분산 제어 팀

- 민주당 팀
- 결정은 합의를 통해 이루어진다.
- 서로의 작업을 검토하세요.
- 이는 사기를 높이고 이직률을 낮추는 데 도움이 됩니다.
- 장기 프로젝트에 적합합니다.
- 이해하기 어렵고 복잡한 문제에 적합합니다.
- 대규모 팀에는 적합하지 않습니다.

의사소통 패턴:

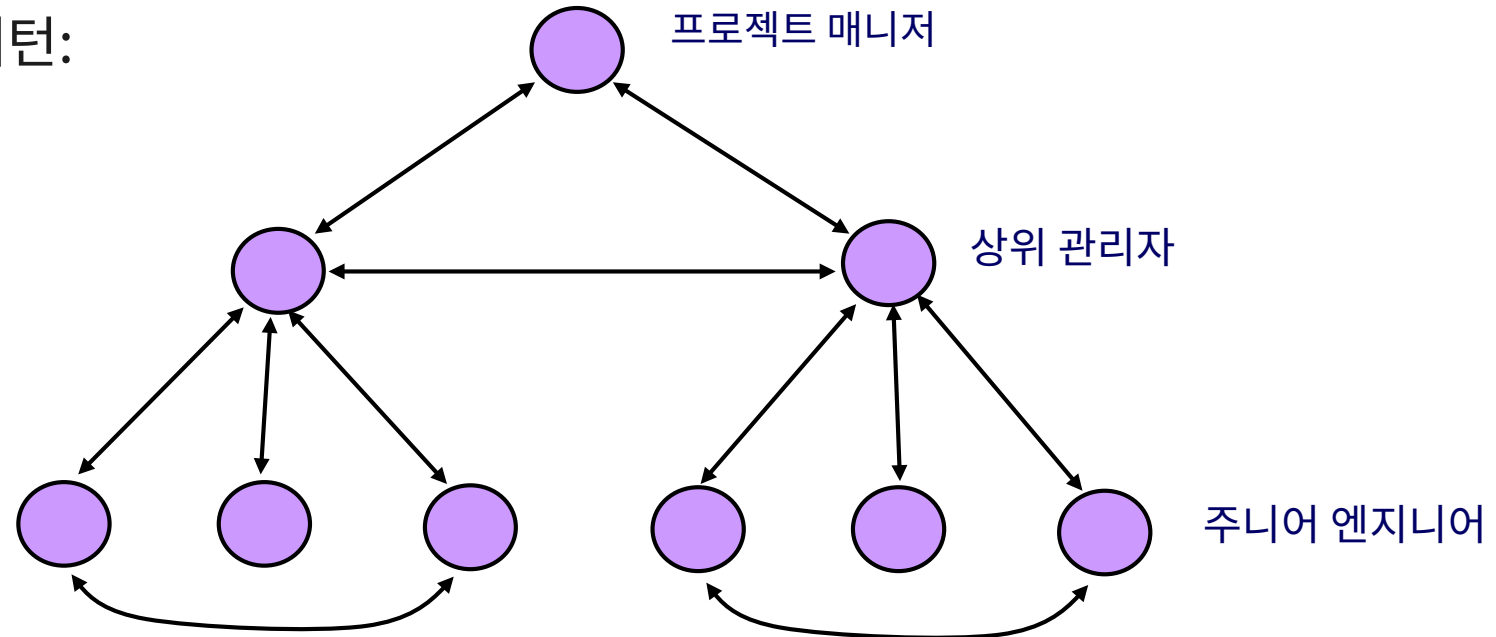


조직 구조 (계속)

혼합 제어 팀

- 계층적 팀
- 중앙집중형 팀과 분산형 팀의 결합.
- 두 가지의 장점을 모두 얻으려는 시도.

의사소통 패턴:

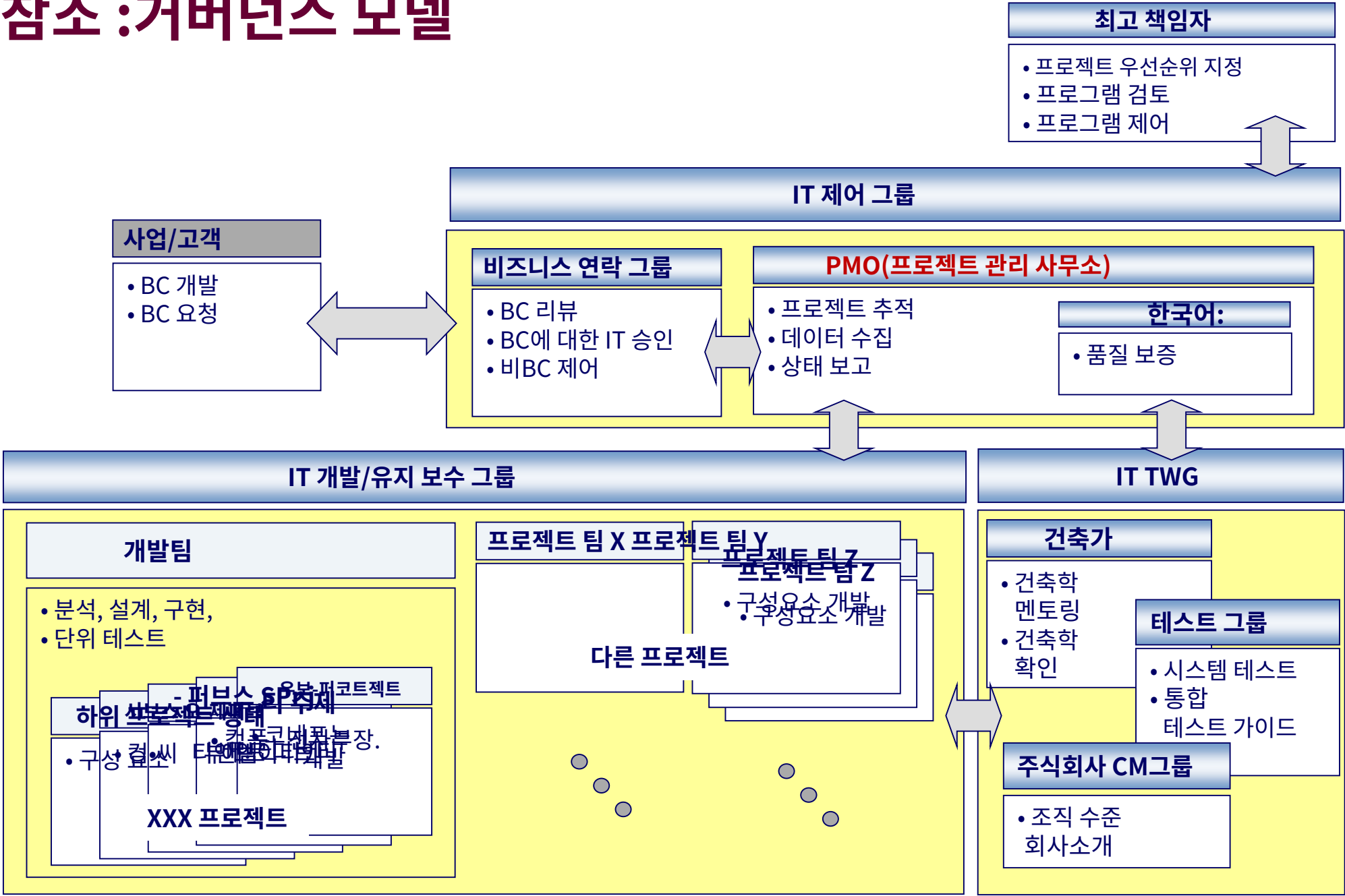


조직 구조 (계속)

팀 조직의 평가

- 모든 업무에 적합한 팀 조직은 없습니다.
- 엔지니어 간의 의사소통이 필요한 경우에는 분산형 제어가 가장 좋습니다.
- 개발 속도가 가장 중요한 목표이며, 문제가 잘 이해되어 있는 경우에는 중앙 집중식 제어가 가장 좋습니다.
- 의사소통의 양을 필요한 것으로 제한하세요.
- 개발 속도 이외의 목표를 고려하세요
 - 수명 주기 비용 절감
 - 감소된 인력 교체
 - 주니어 엔지니어에서 시니어 엔지니어로의 발전
 - 전문 지식과 전문성의 광범위한 보급.

참조 :거버넌스 모델



위험 관리

소프트웨어 엔지니어링의 일반적인 관리 위험:

- 요구사항의 변경.
- 프로젝트에 적합한 사람들이 참여하지 않음.

위험을 줄이는 방법:

- 프로토타입 제작
- 증분 배달
- 모듈식 디자인(변경 사항을 쉽게 수용할 수 있음)

위험 처리 양식(예시)

아이디	위험	비판적인.	발생하다.	완화 계획	대안	답변.
1	직원 이직률	높은	중간	업무량 감소	다중 역할 할당	제홍
2	요구 사항 변경					
	일정 지연					

소프트웨어 엔지니어링 분야의 일반적인 위험

위험 항목	위험 관리 기술
인력 부족	최고 인재 채용; 직무 매칭; 팀 빌딩; 주요 인력 계약; 교차 훈련; 주요 인력 사전 일정 조정
비현실적인 일정과 예산	세부적인 다중 소스 비용 및 일정 추정, 증분 개발, 소프트웨어 재사용, 요구 사항 스크러빙
잘못된 것을 개발하다 소프트웨어 기능	조직 분석; 임무 분석; 운영 개념 수립; 사용자 설문 조사; 프로토타입 제작; 초기 사용자 매뉴얼
금도금	요구 사항 정리, 프로토타입 제작, 비용 편익 분석, 비용 대비 설계
계속되는 스트림 요구 사항	높은 변경 임계값; 정보 은폐; 증분 개발(변경 사항을 이후 증분으로 연기)

그리고 또 다른게 있나요?

(출처, Boehm 1989)

SPMP의 내용

* 문서의 내용은 조직의 표준에 따라 달라집니다.

1. 서론

- 본 문서의 목적
- 프로젝트 개요
- 관련 문서, 용어, 약어

2. 개발 계획

- 리소스 : 인력, 비용
- 작업
- 일정(간테 차트)

3. 조직

- 팀 구조
- 역할 및 책임

4. 기술 관리

- 변화 관리
- 구성 관리
- 기술 관리

5. 품질관리

- 리뷰 방법
- 주기적 검토
- 기타 품질 관리 기술

6. 개발 환경

- 필요한 소프트웨어 및 사양
- 하드웨어 사양
- 공간과 보안
- 곤

7. 성과물

- 문서를 정의하다
- 날짜와 목적지

8. 기타

- 상당한 문제

9. 참고문헌 및 부록

요약 및 토론

프로젝트 관리 단계

-계획→자원의 획득→실행→모니터링

소프트웨어 생산성(또는 크기) 추정

-현지, FP

프로젝트 제어 기술

-WBS, 간트 차트, PERT 차트

팀 조직

프로젝트 위험 관리

효과적인 프로젝트 관리로 개선할 수 있나요?

소프트웨어 품질?

