



Chapter 22. 정리

Chapter 22. 소프트웨어 테스트 전략

현실을 극복하고, 어떤 상황에서도 극복하는 용기

1. 인생이란 원래 공평하지 못하다. 그런 현실에 대하여 불평할 생각하지 말고 받아들여라.
2. 세상은 네 자신이 어떻게 생각하든 상관하지 않는다. 세상이 너희들한테 기대하는 것은 네가 스스로 만족하다고 느끼기 전에 무엇인가를 성취해서 보여줄 것을 기다리고 있다.
3. 대학교육을 받지 않는 상태에서 연봉이 4만 달러가 될 것이라고는 상상도 하지 말라.
4. 학교선생님이 까다롭다고 생각되거든 사회 나와서 직장 상사의 까다로운 맛을 한번 느껴 보라.
5. 햄버거 가게에서 일하는 것을 수치스럽게 생각하지 마라. 너희 할아버지는 그 일을 기회라고 생각하였다.
6. 네 인생을 네가 망치고 있으면서 부모 탓을 하지 마라. 불평만 일삼을 것이 아니라 잘못된 것에서 교훈을 얻어라.
7. 학교는 승자나 패자를 뚜렷이 가리지 않을 지 모른다.
어떤 학교에서는 낙제제도를 아예 없애고 쉽게 가르치고 있다는 것을 잘 안다. 그러나 사회 현실은 이와 다르다는 것을 명심하라.
8. 인생은 학기처럼 구분되어 있지도 않고 여름 방학이란 것은 아예 있지도 않다.
네가 스스로 알아서 하지 않으면 직장에서 가르쳐주지 않는다.
9. TV는 현실이 아니다. 현실에서는 커피를 마셨으면 일을 시작하는 것이 옳다.
10. 공부 밖에 할 줄 모르는 "바보" 한 테 잘 보여라. 사회 나온 다음에는 아마 그 "바보" 밑에서 일하게 될지 모른다





제31장.프로젝트 관리

May. 2018

Young-gon, Kim

ykkim@kpu.ac.kr

Department of Computer Engineering

*K*orea *P*olytechnic *U*niversity



Topics covered

- ◆ 프로젝트 관리 개요
- ◆ 프로젝트 관리 : 사람, 제품, 프로세스, 프로젝트
- ◆ 소프트웨어 프로세스와 프로젝트
- ◆ 프로젝트 관리 활동
- ◆ 프로젝트 계획
- ◆ 프로젝트 스케줄링
- ◆ 위험 관리

1. 프로젝트 관리 개요

◆ 프로젝트 관리

- 계획, 작업 모니터링, 사람들에 대한 제어, 프로세스
- 소프트웨어 초기의 개념으로부터 완전히 작동하여 배포되는 기간 중에 일어날 수 있는 사건 까지를 포함.

◆ 관리의 범위

- 4P : People, Product, Process, Project
- 관리자
 - 인간 의 노력이 집중(소프트웨어공학 업무) 중요성 간과
=> 프로세스관리에 실패
 - 제품 개발의 초기에 이해관계자들과 포괄적인 의사소통을 하지 않는다면
=> 해결책은 좋으나, 엉뚱한 문제에 해결 제시할 위험
 - 프로세스에 관심 부족(우수한 기술적 방법과 툴 사용)
=> 성과 획득 어려움 위험
 - 견실한 계획 없이 프로젝트 착수
=> 프로세스가 위태로움.

1. 프로젝트 관리 개요

◆ 관리의 범위

● 1) 인간

➤ 인적 요인(People factor) 중요

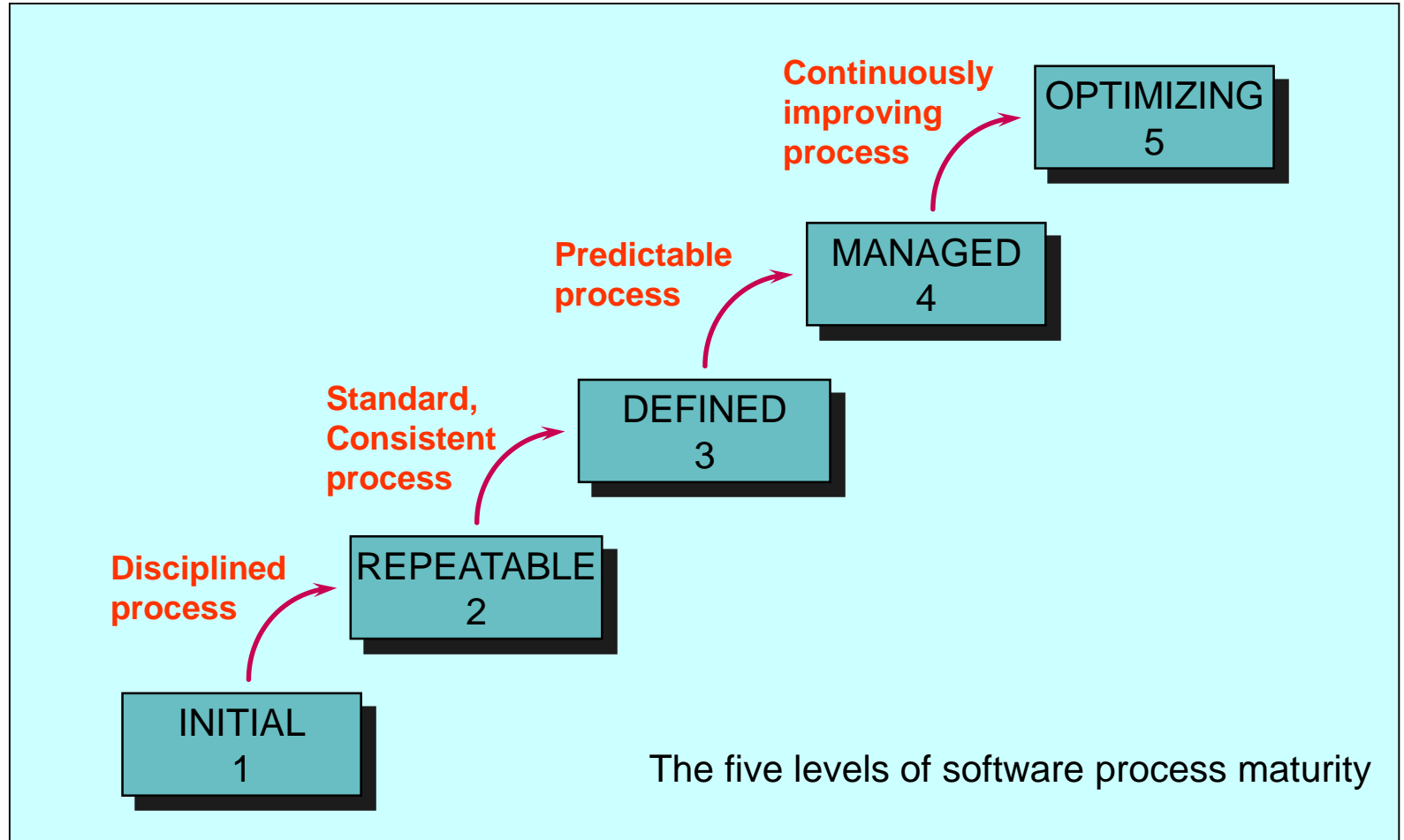
- 모든 기관 : 비즈니스 전략 목표 달성 하기 위해 작업 인력에게 **매력**을 느끼게 하고, 그들을 **개발**, **동기 부여**, **조직화**, **작업 능력**을 지속적으로 향상

➤ 인적 역량 성숙 모델(People-CMM)

- 소프트웨어 인력을 위한 중요한 **실무** 분야 정의
 - . 사람의 **배정**, **의사소통과 협동**, 작업 **환경**, **훈련**, **보상**, **수행능력** 분석 및 개발, **경력** 개발, **작업진단** 개발, **팀/문화** 개발
- 높은 인적 역량 성숙도 **성취**한 조직
 - . 소프트웨어 프로젝트 관리 업무를 **효과적으로 수행** 가능성이 높음.
- 조직이 성숙한 소프트웨어 프로세스 생성 지침
 - . 소프트웨어 역량 성숙 모델 : **통합(SCMM I)**.

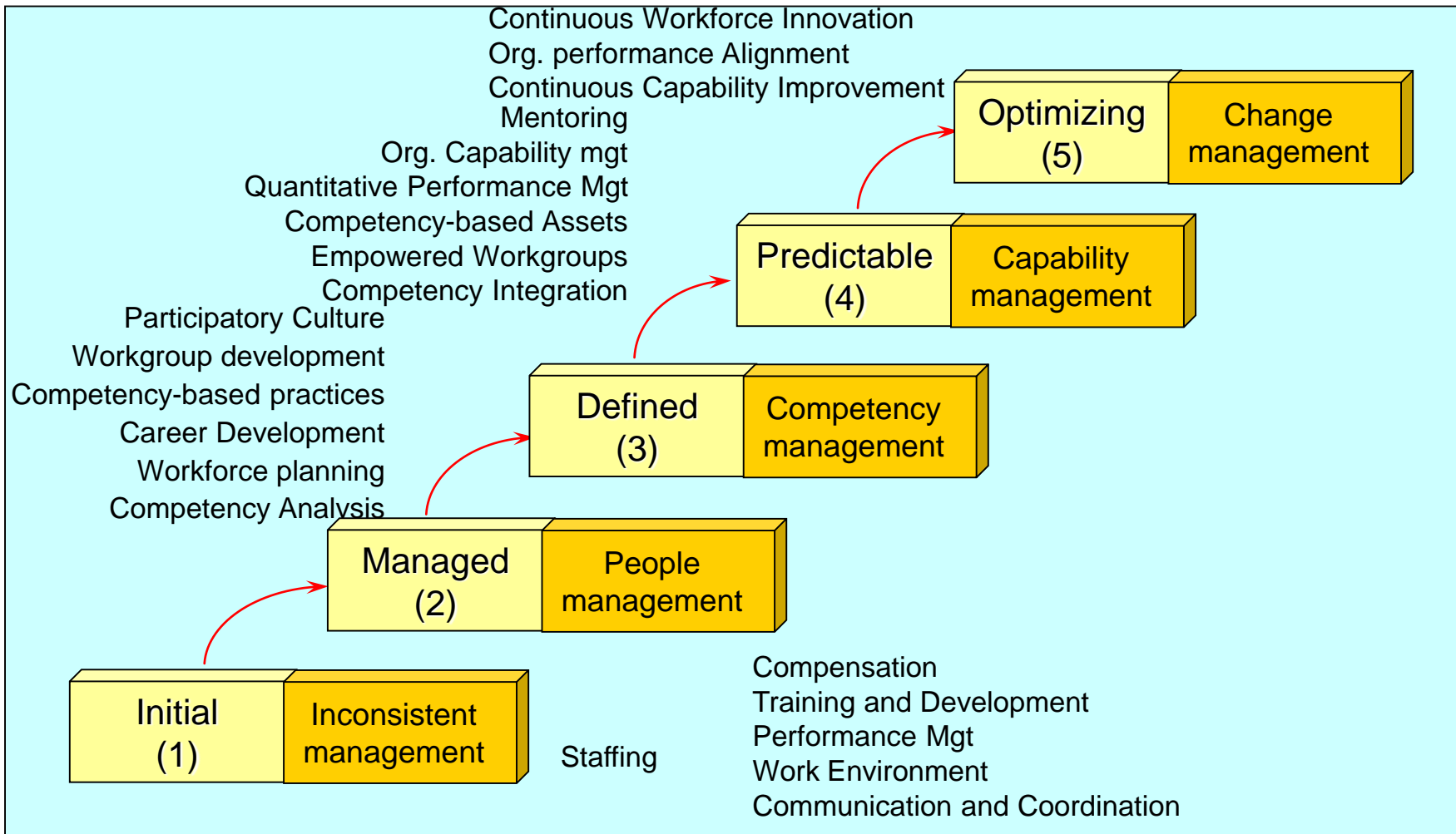
1. 프로젝트 관리 개요

◆ CMM(Capability Maturity Model)



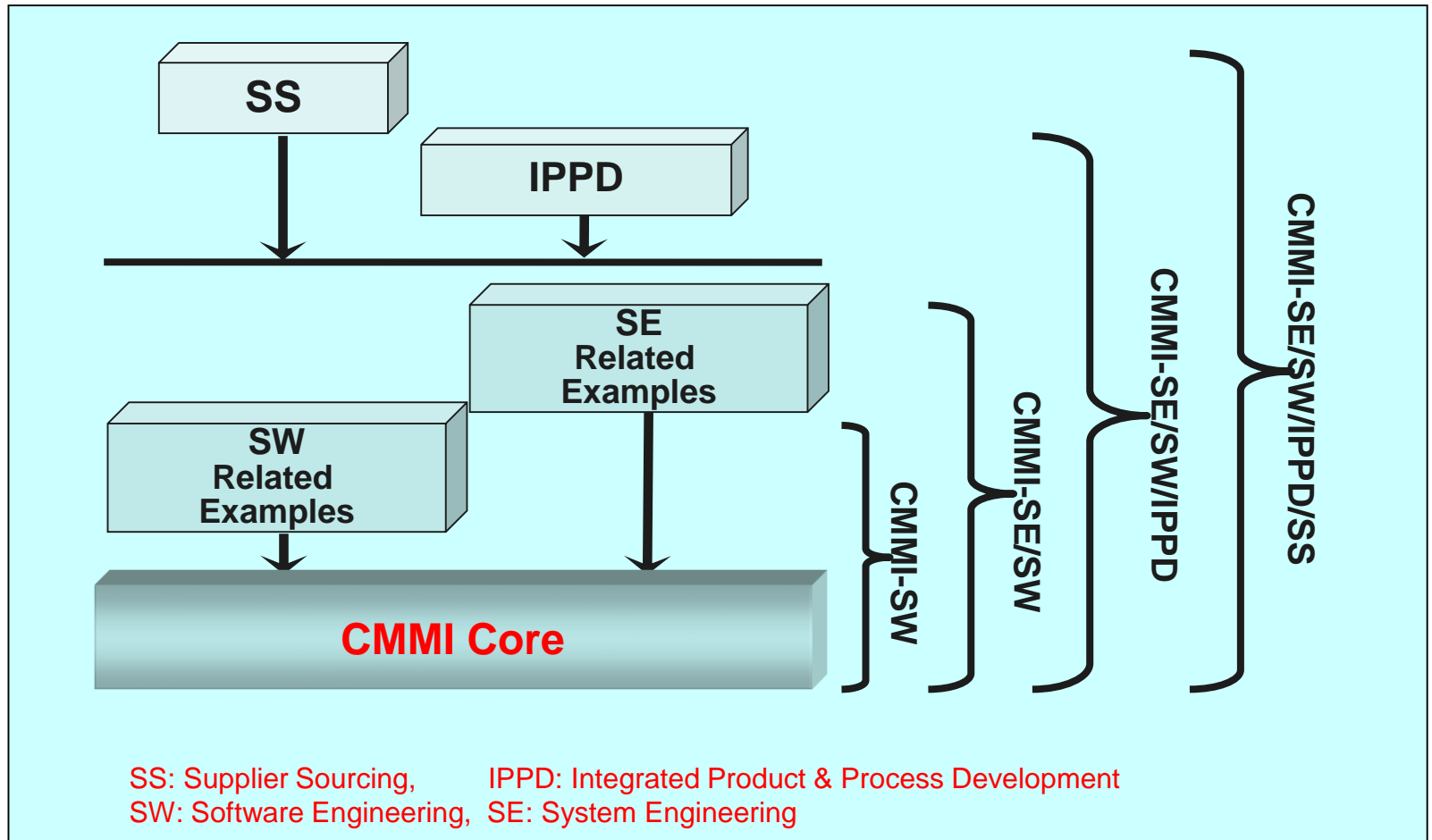
1. 프로젝트 관리 개요

◆ P-CMM(People CMM)



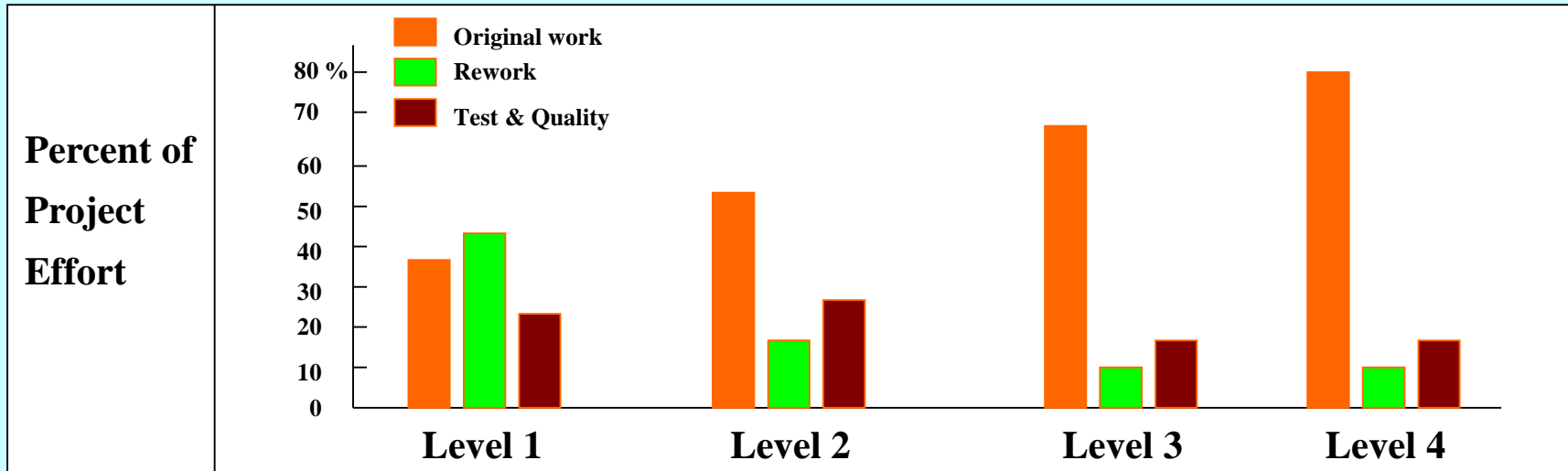
1. 프로젝트 관리 개요

◆ CMMI (CMM Integration)



1. 프로젝트 관리 개요

◆ 효과 : CMM 레벨 향상에 따른 프로젝트 작업 현황



criteria	Level 1-> 2	Level 2-> 3	Level 3->4
Reduce defects	12 %	40 %	85 %
Reduce cycle time	10 %	38 %	63 %
Reduce cost	8 %	35 %	75 %
Schedule variance	145 %	24 %	15 %

1. 프로젝트 관리 개요

◆ 관리의 범위

● 2) 제품

- 프로젝트가 계획 되기 전 업무
 - 제품의 목적과 범위가 수립
 - 해결책 대안 고려
 - 기술적, 관리적 제한조건 파악
- 프로젝트 계획 전 업무에 대한 상세한 정보가 없을 경우 : 정의 불가능
 - 합리적인(정확한) 비용 추정
 - 위험요소의 효과적인 평가
 - 관리 가능한 프로젝트의 일정(현실적인 프로젝트 작업의 분화 또는 진행 상황 징후)
- 제품의 범위
 - 기본적인 데이터, 기능 그리고 제품을 특징짓는 행위
 - 특성들의 정량적인 방법으로 범위 정의
- 제품의 목적과 범위가 확실 : 해결책에 대한 대안 고려
 - 대안
 - . 납품 시한, 예산상의 제약, 개인적 능력, 기술적 상호작용
 - . 많은 요인에 의한 제한조건하에서 “최선의” 해결방안 선택 가능.

1. 프로젝트 관리 개요

◆ 관리의 범위

● 3) 프로세스

➤ 소프트웨어 프로세스

- 소프트웨어 개발을 위한 **종합적인 계획**을 수립할 수 있는 **프레임워크** 제공
- 소프트웨어 **프로젝트 특성**들과 프로젝트 **팀의 요구사항**들을 프레임워크 작업에 적응
 - . **Task set** : 작업, 일정표, 산출물, 품질 보증 요건
- 프로세스 모델에 추가사항 : **보호 행위**
 - **소프트웨어 품질 보증, 소프트웨어 형상관리 및 측정.**

● 4) 프로젝트

- **계획되고 제어된** 소프트웨어 프로젝트 : **복잡도를 관리**할 수 있는 유일한 방법.
- 소프트웨어 프로젝트 성공/실패 율 : 250개 대형 프로젝트(1998~2004)
 - **25개** 프로젝트 : **일정,비용, 품질목표 달성**[10 %]
 - **50개** 정도 프로젝트 : **35% 미만 일정 지연, 비용 초과**[20 %]
 - **175개** 프로젝트 : **크게 지연/초과, 미완성된 채 종결**[70 %]
- 프로젝트 실패 피하는 방법
 - 프로젝트 관리자/엔지니어 **경고 신호** 피해야 함
 - 프로젝트를 계획, 관찰, 제어할 수 있는 **상식적인 방법** 개발.

2. 프로젝트 관리- 사람

- ◆ 이해관계자 : 소프트웨어 프로세스에 속하는 5가지 범주
 - 선임관리자
 - 비즈니스 문제 정의 : 프로젝트에 막대한 영향을 미침.
 - 프로젝트 관리자
 - 계획, 동기 부여, 조직, 제어 : 소프트웨어 작업을 담당할 실무자들 .
 - 실무자
 - 전문적인 기술 제공 : 제품이나 응용프로그램 제작에 필요.
 - 고객
 - 요구사항 및 관심사 상술 : 제품의 요구사항, 다른 이해 관계자들의 결과물 관심사.
 - 최종 사용자
 - 소프트웨어와 상호관계를 갖음 : 제품이 실제적 사용.

2. 프로젝트 관리- 사람

◆ 팀 리더

● 리더의 역할

- 동기 부여 : 기술자들이 최대한의 능력을 발휘하도록 격려하는 능력(“밀어주고 끌어주며”)
- 조직화 : 초기 개념이 최종 제품이 되도록 기존의 프로세스 모델화 (신 프로세스 고안) 능력
- 아이디어 혁신 : 사람들이 창의적일 수 있도록, 창의적이라고 느낄 수 있도록 격려하는 능력

● 성공적인 프로젝트 리더

- 해결할 문제 이해, 아이디어 흐름 관리, 품질이 중요/양보할 수 없다는 것을 팀원 알리는데 전력

● 유능한 관리자 정의하는 특성

➢ 문제해결

- 가장 관련성 깊은 기술적/조직적 문제 진단, 체계적인 해결책 조직화, 실무자들이 해결책 개발할 수 있도록 동기부여, 새로운 상황에 과거 프로젝트 교훈 적용, 문제해결의 초기 해결 효과 없을 시 방향을 바꿀 수 있는 유연성

➢ 관리자로서의 주체성

- 책임을 져야 함, : 감독권 행사, 훌륭한 기술자들이 자신의 직감을 따를 수 있도록 보장

➢ 성취감

- 팀의 생산성 최적화 솔선하는 사람/공적이 있는 사람에게 보상, 계획된 위험 부담에 벌하지 않는다는 것을 자신의 행동으로 보여야 함.

➢ 영향력과 팀 구성

- 사람의 마음을 “읽어” 수 있고, 사람의 말/비언어적인 신호 이해, 신호 요구에 반응, 스트레스에 자신을 제어.

2. 프로젝트 관리- 사람

◆ 소프트웨어 팀

- “최선의” 팀 의존 요소
 - 조직 관리 방식, 팀에 속한 사람의 수와 기술 수준, 문제의 전반적인 난이성에 좌우
- 소프트웨어 공학팀의 구조를 계획 시 고려할 7가지 프로젝트 요소
 - 해결할 문제 난이도, 프로그램 결과의 코드 라인 수/기능점수의 “크기”, 팀이 함께 일하는 기간(팀 수명), 문제가 모듈화되는 정도, 개발할 시스템에 요구되는 품질 및 신뢰도, 납품 기일에 대한 엄정성, 프로젝트를 위해 요구되는 의사소통의 수준
- 소프트웨어 공학팀을 위한 4가지 “조직의 방법론”
 - 폐쇄적인 방법론 : 팀 리더 중심
 - 전통적인 직무 체계에 따른 구조,
 - 과거의 작업과 유사한 개발에 효과적, 별로 혁신적이지 않음
 - 무작위적인 방법론 : 혁신적 무질서의 표현에 어울리는 창의적 독립적인 팀
 - 느슨한 구조, 팀원의 개인적인 창의성에 의존,
 - 혁신/기술적 돌파구 필요 경우 탁월, “규율이 중요한 작업”어려움.
 - 개방적인 방법론
 - 개방적인 방법론의 특성인 의사소통 및 공감대 기반의 의사결정에 비중을 두어 협동하여 수행
 - 복잡한 문제를 해결하는 도움, 효율적이지는 않을 수 있음.
 - 동시적인 방법론
 - 문제가 자연히 구획화되는 것이라 기대하여 팀원들이 서로 의사소통을 거의 하지 않은 채 문제의 부분들에 대해 작업을 수행할 수 있음.

2.프로젝트 관리- 사람

◆ 소프트웨어 팀

- 성능이 탁월한 팀을 만드는 방법
 - 팀원 들은 서로 신뢰, 기술 배분이 문제에 적절, 팀의 단결을 이루기 위해 독단적인 사람 제외
- 프로젝트 관리자의 목적 : 단결된 팀 구성
 - 융합된 팀 : 각자를 더한 것보다 전체가 더 커질 수 있게 한 무리의 사람들이 강하게 결합
 - 보통 팀원보다 생산성이 높고, 동기부여가 잘 됨
 - 공유 : 하나의 목표, 공동의 문화, 정예 의식
- “독소있는 팀(team toxicity)” 환경을 조장하는 5가지 요인
 - 격양된 분위기
 - 회피 방법 : 필요한 모든 정보 접근 확인, 일단 정의되면 특별한 경우외 목표와 목적을 수정하지 말아야 함.
 - 팀원 간에 알력의 원인이 되는 높은 요구불만
 - 회피 방법 : 가능한 많은 의사결정권 부여
 - “단편적이거나 졸렬하게 조정된” 소프트웨어 프로세스
 - 회피 방법 : 구축될 제품과 업무 수행할 사람을 이해, 팀이 프로세스 모델을 선택하게 함.
 - 불분명하게 정의된 소프트웨어 팀 내에서의 역할
 - 회피 방법 : 팀이 직접 책무에 대한 기법(기술적 검토) 수립, 팀원이 수행한 업무 실패 시 일련의 정정안이 정의
 - “지속적이고 반복적인 실패의 조짐”
 - 회피방법 : 팀 기반의 피드백 및 문제 해결 기술 확립.

2. 프로젝트 관리- 사람

◆ 공동 작업과 의사소통의 문제들

- 소프트웨어 프로젝트가 어려움을 겪는 이유
 - 많은 개발 노력은 **규모**가 커서 : **복잡**해지고, **혼돈**이 일어나고, 팀원들이 **공동으로 작업**하는데 큰 어려움 발생
 - **불 확실성**이 만연 : 계속적인 **변경**으로 프로젝트 팀이 방해
 - **상호 운용성** : 많은 시스템에 있어 매우 **중심적인 특징**
- **최신 소프트웨어**의 특성 : **규모, 불 확실성, 상호운용성** -> 현실
 - **효과적 처리** : 작업을 하는 사람들이 **공동으로 작업**을 할 수 있는 효과적인 방법 확립
 - 팀원들 간의 그리고 여러팀 간의 **공식적/비공식적 의사소통**이 이루어 질 수 있는 기법 수립
 - 1) **공식적인 의사소통**
 - **문서화, 조직화**된 회의
 - 비교적 서로 **영향**을 주지 않으면서 **개인적이지 않은** 의사소통 경로 통해 이루어짐
 - 2) **비공식적인 의사소통**
 - **개인적**
 - 팀원들은 **특별한 주제**에 관해 의견 공유, 문제가 야기될 경우에 **도움** 요청
 - **매일 서로** 의견 공유.

2. 프로젝트 관리 - 제품

◆ 소프트웨어의 범위

- 소프트웨어 범위 결정 : 결정을 위한 질문

- 맥락(Context)

- 어떻게 제작 중인 소프트웨어가 보다 큰 시스템, 제품, 비즈니스 환경과 적합 될 수 있는가?
- 그러한 맥락에서 어떤 제한조건이 부과되는가?

- 정보 목적

- 고객이 보는 어떤 데이터 객체가 소프트웨어로부터 출력으로서 만들어지는가?
- 어떤 데이터 객체가 입력으로서 요구되는가?

- 기능과 성능

- 소프트웨어의 어떤 기능이 입력 데이터를 출력으로 변환시키는가?
- 특별히 설명되어야 할 성능의 특성이 있는가?

- 프로젝트 범위

- 관리자 및 기술자의 수준에서 명백해야 하고 이해될 수 있어야 함.

- 소프트웨어 범위에 대한 기술 : 경계가 분명해야 함

- 정량적인 데이터 명시적 기술

- . 동시 사용자수, 메일 주소 목록, 허용되는 최장 응답시간

- 제한 조건 및 한계를 주지 시켜야 함

- . 제품의 비용이 메모리의 크기를 제한

- 완화 요인들이 설명되어야 함.

- . 요망되는 알고리즘이 충분히 이해되었으며, Java로 작성 가능함.

2. 프로젝트 관리- 제품

◆ 문제의 분해

- 분할, 문제의 세분화/분해 : 소프트웨어 요구사항 분석의 핵심적 행위
- 분해는 2가지 영역에서 적용
 - 반드시 제공되어야 할 기능성 및 내용
 - 그들을 제공하기 위해 사용될 프로세스
- 분할 정복(divide-and-conquer) 전략
 - 복잡한 문제를 접했을 때 사용
 - 복잡한 문제를 가능한 작은 문제들로 분할
 - 프로젝트를 계획하기 시작할 때 적용되는 전략
 - 상세한 정보 정보 제공할 수 있게 다듬어 짐
 - 범위에 대한 서술에 설명되는 소프트웨어의 기능들을 평가, 추정을 시작 전
 - 비용과 일정에 대한 추정 : 모두가 기능에 유래 의존
 - 컴포넌트 분해
 - 주요 내용 및 데이터 객체들은 소프트웨어가 생성할 정보를 무리 없이 이해.

2. 프로젝트 관리- 프로세스

◆ 팀에 적합한 프로세스 모델 선정

● 팀에서 프로세스 모델 결정

- 제품을 요구한 **고객**과 그 일을 할 **사람들**
- 제품의 **특성** 그 자체
- 소프트웨어 팀의 작업 환경에 가장 적합한 **모델인가를 결정**

● 프로젝트 계획 수립 프로세스

- **프로세서 모델**이 선정
- 팀은 **예비 프로젝트 계획** 정의 <- 모델 선정후 : 프로세스의 프레임워크 작업들을 기반
- **예비 계획** 수립
- **프로세스의 분해**가 시작
- **완성된 계획 수립** : 프레임워크 작업을 수행하는데 수반되는 작업 반영.

2. 프로젝트 관리- 프로세스

◆ 프로세스 분해

- 프로젝트 관리자가 “ 이 프레임워크 작업을 어떻게 완수할 것인가?”로 시작
- 소규모이며 상대적으로 단순한 프로젝트 : 의사소통 작업 분해 표본
 - 1) 명확히 해야 할 문제들의 목록을 작성
 - 2) 명확히 해야 할 문제들을 이해 관계자들에게 설명하기 위해 회합을 가짐
 - 3) 범위를 정하는 문서를 함께 작성
 - 4) 모든 이해관계자들과 범위를 정한 문서를 검토
 - 5) 필요하다면 범위를 정한 문서를 수정.

2. 프로젝트 관리- 프로세스

◆ 프로세스 분해

- 대규모이며 상대적으로 복잡한 프로젝트 : 의사소통 작업 분해 표본

- 1) 고객의 요구를 재검토
- 2) 공식적이고 일을 촉진하는 모든 이해관계자들과의 회의를 계획하고 일정을 정함
- 3) 제안된 해결책과 기존의 방침을 명세하기 위한 연구를 수행
- 4) 공식 회의를 위해 작업 문서와 안건을 준비
- 5) 회의를 개최
- 6) 소프트웨어의 데이터, 기능 및 행위의 특징을 반영하는 소규모 명세서를 함께 작성
대안으로, 사용자의 관점에서 본 소프트웨어를 기술하는 유스케이스를 작성
- 7) 각 소규모 명세서나 유스케이스의 정확성, 일관성이 있으며, 그리고 애매모호성이 없는지 검토
- 8) 소규모 명세서를 모아 정리하여 범위를 정하는 문서를 작성
- 9) 범위를 정하는 문서, 또는 유스케이스의 집합을 관심 있는 사람 모두와 함께 검토
- 10) 필요하다면 범위를 정하는 문서, 또는 유스케이스들을 수정.

2. 프로젝트 관리 - 프로젝트

◆ 신뢰의 법칙(Rule of Credibility) : 힘든 프로젝트 토의할 때 인용

- **90-90 법칙** : “시스템의 처음 90% 부분이 전체 할당된 시간과 노력의 90 차지하고, 마지막 10% 역시 전체에 할당된 시간과 노력의 90%를 또 차지한다 ”
 - 고객 필요로 하는 것 **이해** 부족,
 - **제품 범위** 불충분하게 정의,
 - **변경**을 서투르게 관리,
 - **채택 기술** 변화,
 - **비즈니스** 사항 변경,
 - **마감기일** 비현실적 정의,
 - **새로운 시스템**에 대한 반감,
 - **스폰스** 이탈,
 - 프로젝트 팀 꼭 **필요한 기술** 미확보,
 - 개발자 실수에 **교훈**을 얻지 못한 경우 <- 문제점.

2. 프로젝트 관리 - 프로젝트

- ◆ 소프트웨어 프로젝트 5가지 **상식적 접근법 : 문제점 회피 방법**(관리자)
 - 1) **시작을 잘하라(start on the right foot)**
 - 프로젝트 관련자 : 현실적인 목표, 기대 확립, 팀에게 자율성, 권한, 필요기술 부여
 - 2) **여세를 유지하라(Maintain momentum)**
 - 의욕저하 최소화 : 장려책 마련, 품질 강조, 선임관리자 팀에 방해 되지 않도록 전력
 - 3) **진행 상황을 추적하라(Track progress)**
 - 품질보증의 일환으로 산출물 생산, 승인->진행상황 추적, 측정->진행상황과 개발 작업 평균과 비교
 - 4) **현명한 결정을 하라(Make smart decisions)**
 - “항상 단순하게 결정”, 기존 컴포넌트/패턴 사용 결정, 표준화된 인터페이스 사용, 위험관리 기법
 - 5) **사후분석을 시행하라(Conduct a postmortem analysis)**
 - 교훈을 도출할수 있는 일관성 기법 수립, 계획/실제 일정 평가, 프로젝트 척도수집, 얻은지식 문서화.

3. 소프트웨어 프로세스와 프로젝트

◆ 프로세스와 프로젝트

- 단순한 프로젝트에 대해 단순한 프로젝트 계획을 수립
 - 프로젝트의 목적, 이정표, 일정, 책임, 관리 및 기술방법, 필요 자원

◆ 5W2H 원칙 : 중요한 프로젝트 특성과 결과로서 프로젝트 계획 정의

- 왜(Why) 시스템은 개발되고 있는가? : 비즈니스 이유의 타당성 평가
 - 비즈니스의 목적이 사람, 시간, 돈의 비용을 정당화하는가?
- 무엇(What) 을 할 것인가? : 프로젝트가 정의되기 위해 요구되는 일련의 작업들
- 언제(When) 할 것인가? : 프로젝트 일정 수립
 - 팀이 언제 프로젝트 작업 수행할 것인가, 언제 중요한 milestone에 도달 할 것인가 파악
- 누가(Who) 어떤 기능에 대해 책임을 있는가? : 팀원 각자의 역할과 책임정의
- 조직 내에서 그들은 어디(Where) 에 위치하는가?
 - 이해관계자에 역할과 책임 : 실무자, 고객, 사용자, 다른 이해관계자
- 어떻게(How) 기술적으로 관리적으로 임무를 달성할 수 있는가?
 - 제품 범위가 수립 후 : 프로젝트를 위한 관리 및 기술적 전략 정의
- 각 자원들은 얼마만큼(How much) 필요한가?
 - 5W1H 질문의 답을 기준으로 추정 수립하여 도출.



4. 프로젝트 관리 활동

◆ 관리자가 책임질 관리항목

- **Proposal writing** (제안서 작성)
 - 프로젝트 목표/수행방법 기술, 비용/일정 추정치, 특정 팀 계약 합리화
- **Project planning and scheduling** (프로젝트 계획 수립 및 일정 관리)
 - 제품생성 활동, 기준, 이정표, 산출물 식별
- **Project costing** (프로젝트 비용 산정)
 - 요구되는 자원 추정
- **Project monitoring and reviews** (프로젝트 감시 및 검토)
 - 진행상황 추적, 계획 대비 진도와 비용비교, 자원이 조직 목표와 일치 검사, 변경
- **Personnel selection and evaluation** (인력 선발 및 평가)
 - 숙련된 인력 참여, 제약조건(Expert, 인력 양성)에서 인력 선발
- **Report writing and presentation** (보고서 작성 및 발표)
 - 상세 프로젝트 보고서-> 추상적이고 간단한 요약 보고서 작성, 효율적 표현.

5. 프로젝트 계획

◆ 프로젝트 계획서에 포함되어야 할 내용

- 프로젝트 범위 및 프로젝트 스케줄
- 프로젝트 팀 구성
- 제안된 시스템의 기술적 기술(technical description)
- 프로젝트 표준안, 절차, 제안된 기술 및 도구
- 품질 보증 계획, 형상 관리 계획
- 문서화 계획(documentation plan)
- 데이터 관리 계획(data management plan)
- 자원 관리 계획(resource management plan)
- 테스트 계획
- 교육 계획
- 보안 계획
- 위험관리 계획
- 유지보수 계획.

5.프로젝트 계획

◆ 프로젝트 계획서

1. 개요

- 1.1 프로젝트 개요
- 1.2 프로젝트의 산출물
- 1.3 정의, 약어

2. 자원 및 일정 예측

- 2.1 자원
 - 가. 인력
 - 나. 비용

2.2 일정

3. 조직 구성 및 인력 배치

- 3.1 조직 구성
- 3.2 직무 기술

4. WBS

5. 기술 관리 방법

- 5.1 변경 관리
- 5.2 위험 관리
- 5.3 비용 및 진도 관리
- 5.4 문제점 해결 방안

6. 표준 및 개발 절차

- 6.1 개발 방법론

7. 검토 회의

- 7.1 검토회 일정
- 7.2 검토회 진행 방법
- 7.3 검토회 후속 조치

8. 개발 환경

9. 성능 시험 방법

10. 문서화

11. 유지보수

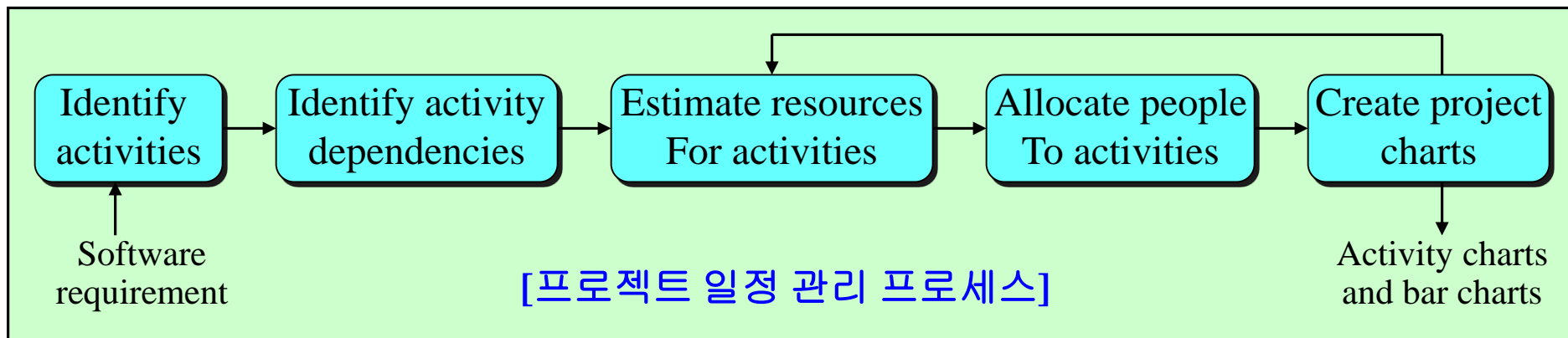
12. 설치, 인수

13. 참고문헌 및 부록

6. 프로젝트 스케줄링

◆ 프로젝트 일정 관리

- 프로젝트의 총 업무를 각 활동 별로 분리하여 각 활동을 완료하는 데 소요되는 시간을 결정
- 특정 업무 지연으로 전체 업무 지연되는 상황 발생되지 않게 작성
- 세분화 : 주단위로 작성, 활동 (8~10주)
- 일정 추정 시 낙관적 예측 금물 : 병과,이직,고장,납품 지연,고도 기술
- 자원 추정 : 사람, 시스템 자원, 특수한 하드웨어, 경비
- 추정치 포함 : 일반 추정치(30%) + **미예상 추정치(20%)**.



6. 프로젝트 스케줄링

◆ Bar charts and activity networks

- 프로젝트 일정 계획을 Graphical notions
 - **바 차트** : 각 활동을 책임지는 사람을 나타내고, 시작/끝 시간을 표현
 - **액티비티 네트워크** : 프로젝트를 구성하는 다른 활동 사이 관계 표현

◆ Activity networks

- 활동간의 종속관계, 추정 시간이 주어지면 -> **활동 순서 정의**
- 사각형(활동), 모서리가 둥근 사각형(이정표와 산출물)
날짜(활동의 시작일 : 영국식 표현[d/m/y])
- 차트 읽는 순서 : 좌 → 우, 상 → 하
- **임계 경로(critical path)** : 가장 긴 경로(프로젝트 소요 최소시간)
 - 임계 경로 길이 최소화 되도록 설계 재수정, 병렬업무 처리.

6. 프로젝트 스케줄링

◆ Task durations and dependencies

Activity	Duration (days)	Dependencies
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)

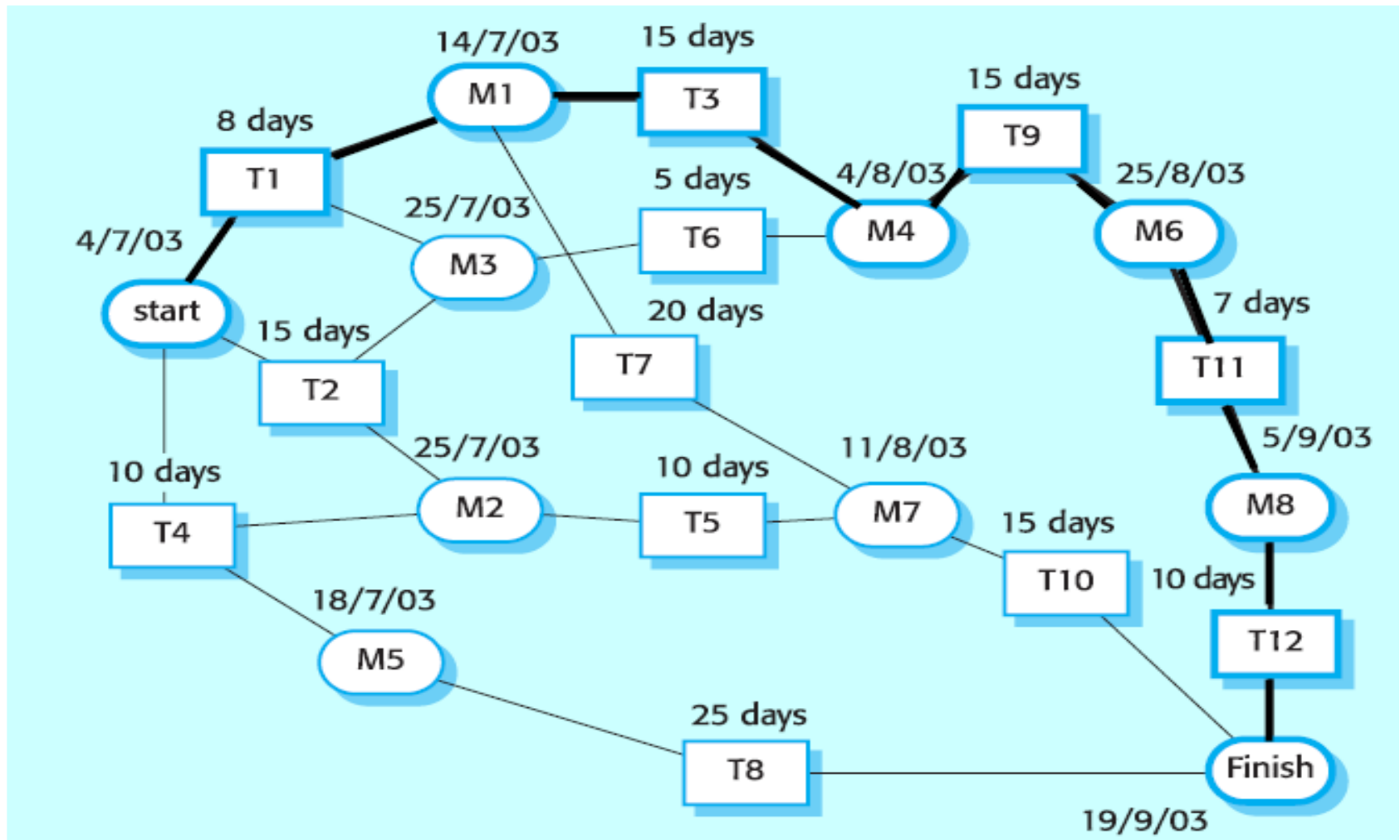
6. 프로젝트 스케줄링

◆ Task durations and dependencies

	Activity	Duration(week)	Dependencies
T1.	시스템 정의	0.5	-
T2.	프로세스 모델 정의	0.5	-
T3.	기능/비기능 요구사항 정의	2	T1
T4.	요구사항 분석	1	T1, T3
T5.	시스템 설계	2	T1, T3 (M1)
T6.	아키텍처 스타일 선정	0.5	T2
T7.	아키텍처 설계 프로세스 정의 및 설계	0.5	T2, T6
T8.	코딩	4	T1, T5 (M2)
T9.	테스트 케이스 설계	1	T3, T5, T8
T10.	화이트박스 테스트	1	T8, T9 (M3)
T11.	블랙박스 테스트	1	T5, T8
T12.	통합 테스트	1	T1, T8 (M4)

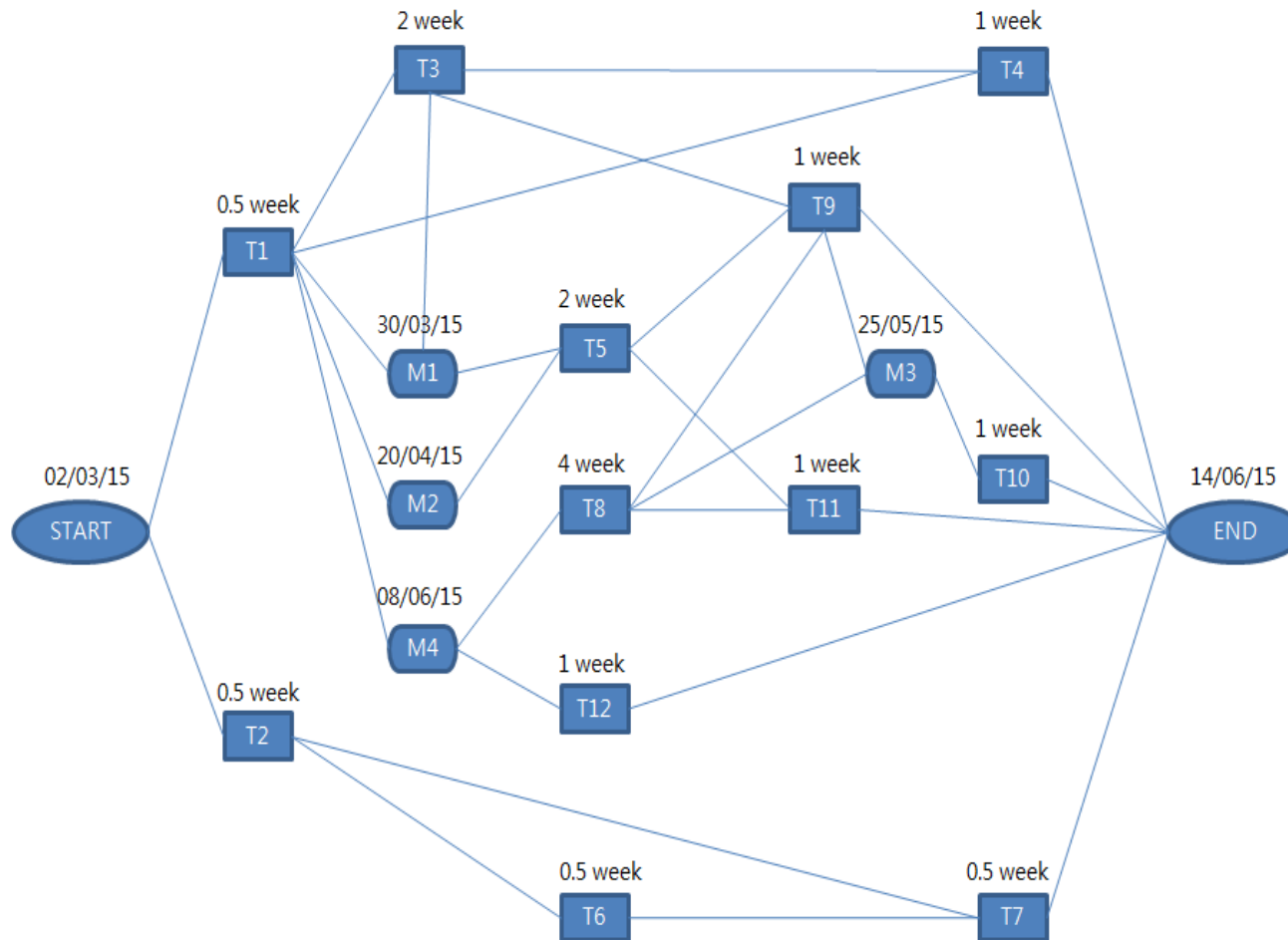
6. 프로젝트 스케줄링

◆ Activity Network



6. 프로젝트 스케줄링

◆ Activity Network





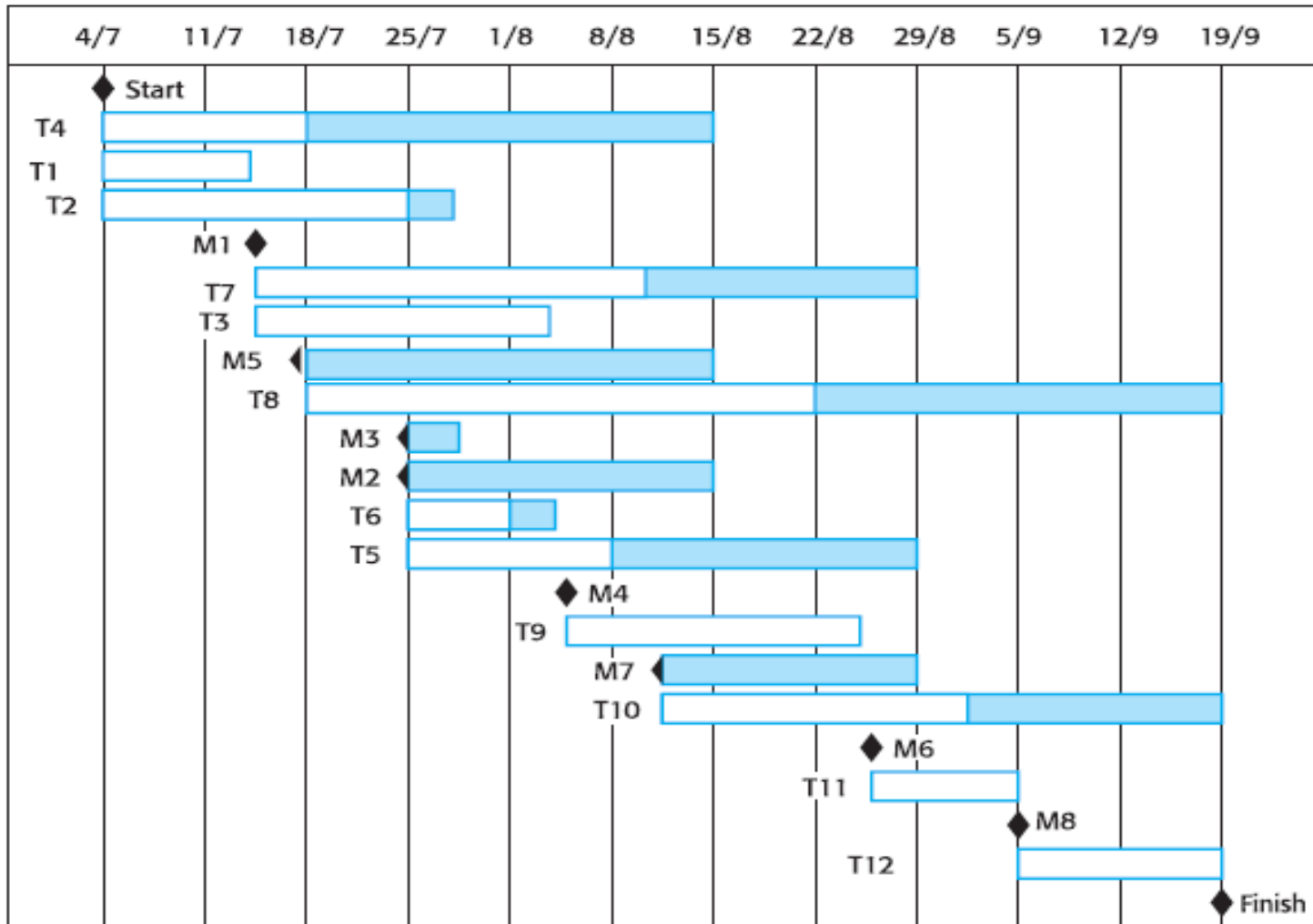
6. 프로젝트 스케줄링

◆ Bar charts(Gantt charts)

- 프로젝트 캘린더와 활동들의 시작일과 종료일 나타냄
- 차트는 좌 → 우로 읽음
- 활동 시작일과 종료일이 분명하게 표현
- 음영바 : 활동 완료시간 융통성
 - 음영시간 까지 임계 경로에 영향을 받지 않는 것
- 일정 고려, 자원 할당, 직원 배정
 - 휴식, 다른 프로젝트 업무 수행, 교육 참여, 다른 활동 참여
 - 전문가를 고려할 있음.

6.프로젝트 스케줄링

◆ Activity Bar Charts



절대경로

T1

T3

T9

T11

T12

종속성

(T1,2)T6

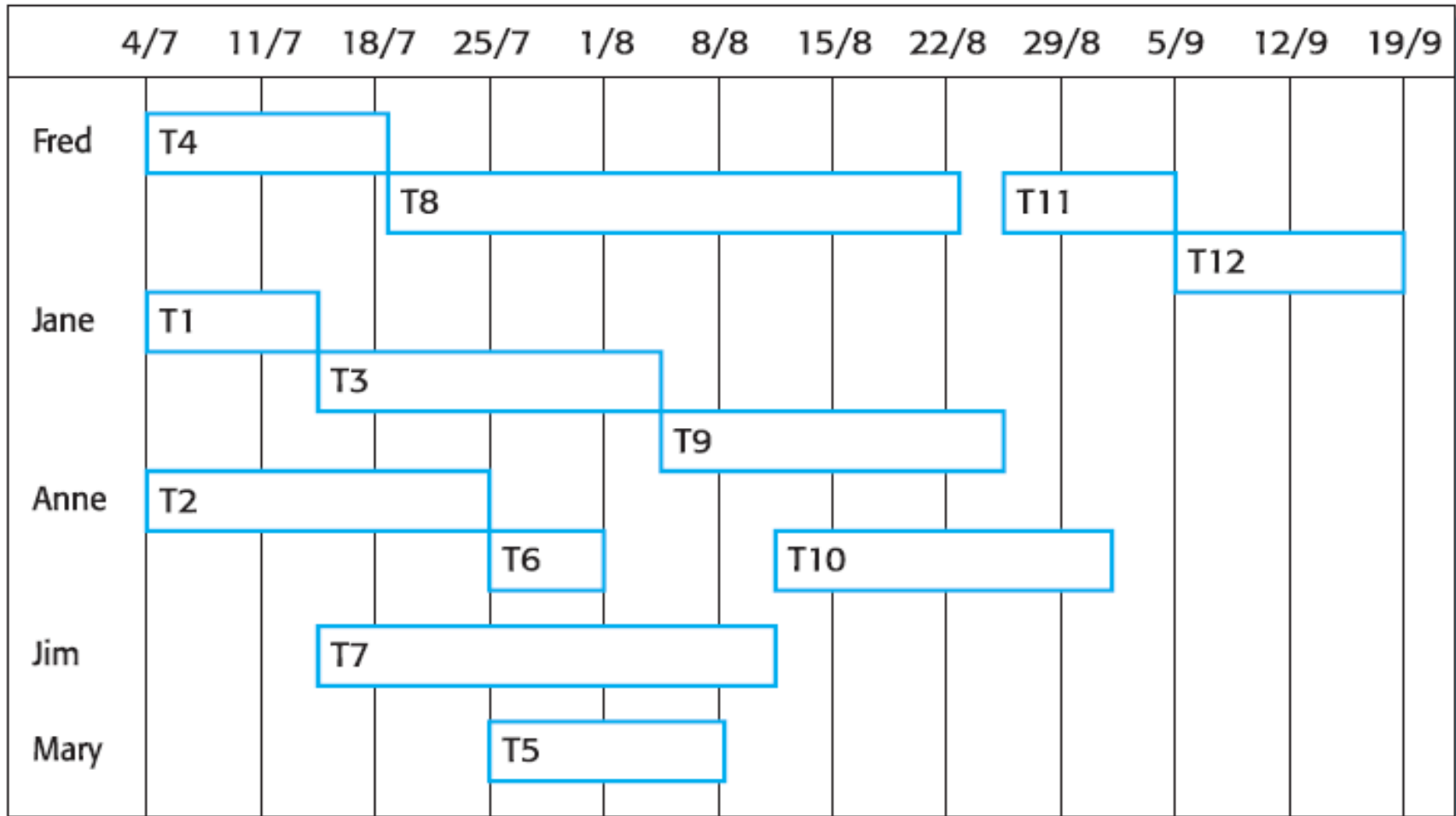
(T2,4)T5

(T3,6)T9

(T5,7)T10

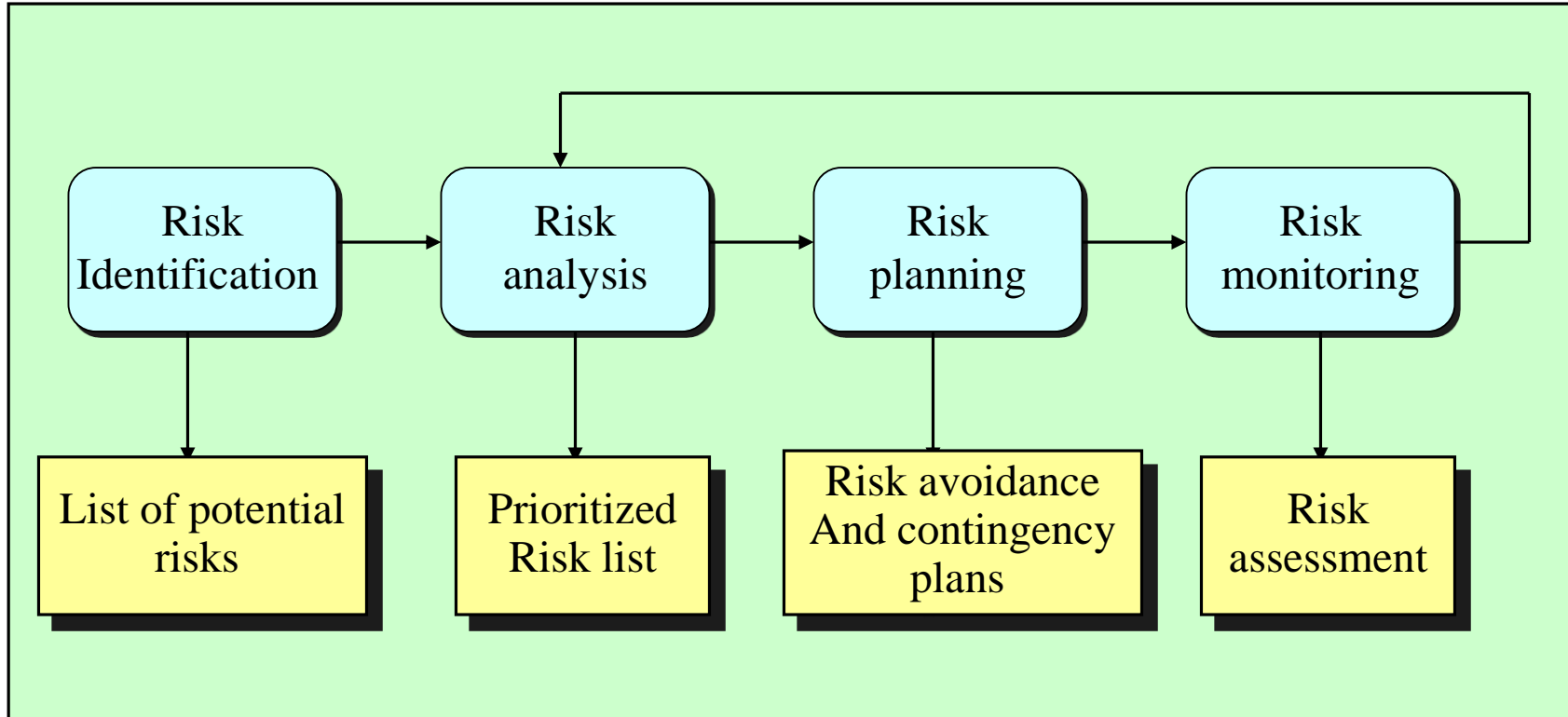
6. 프로젝트 스케줄링

◆ 일정표에 따른 Staff allocation



7.위험 관리

◆ Risk Management Process



7. 위험 관리

◆ Risk identification (위험 식별) 점검 목록

- 기술 위험

- 소프트웨어 혹은 하드웨어 기술로부터 유도된 위험

- 인적 위험

- 개발 팀에 있는 사람과 관련된 위험

- 조직의 위험

- 소프트웨어가 개발되는 조직의 환경과 관련된 위험

- 도구 위험

- 지원 소프트웨어와 CASE 도구로부터 생기는 위험

- 요구사항 위험

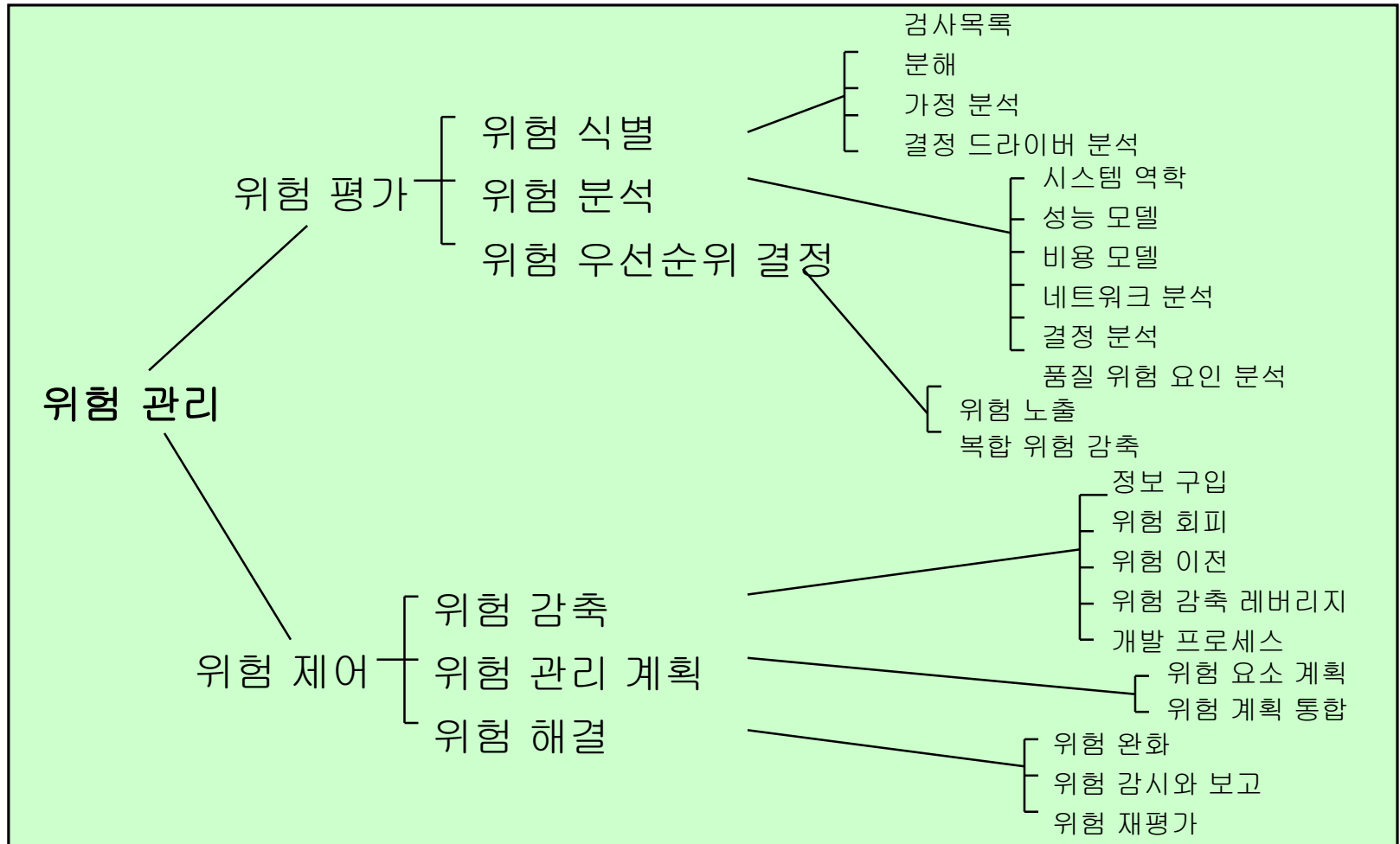
- 요구사항 변경과 관리를 위한 프로세스로부터 생긴 위험

- 추정 위험

- 시스템을 만들기 위해서 필요한 자원에 대한 추정으로부터 생긴 위험.

7.위험 관리

◆ 위험 관리 단계





정리 및 Homework

- 1) 프로젝트 관리 정의
- 2) 프로젝트 관리의 범위와 세부 내용
- 3) 프로젝트 계획서에 포함 내용
- 4) 위험관리 프로세스
- 5) 위험 관리 단계



Project

1장. 프로젝트 개요

- 1.1 프로젝트 제목
- 1.2 선정 이유
- 1.3 팀 운영 방법

2장 시스템 정의

- 2.1 시스템 간략한 설명
- 2.2 유사 사례 간략한 설명

3장 프로세스 모델

- 3.1 규범적인 프로세스 모델 선정 및 이유
- 3.2 특수한 프로세스 모델 선정 및 이유

4장. 실무 가이드 원칙

- 4.1 각 프레임워크 원칙에서 중요한 3 개 정의
- 4.2 프로젝트 계획 보고서

5장. 요구사항 획득

- 5.1 기능 요구사항과 비기능 요구사항 정의
- 5.2 표준 양식을 사용한 시스템 요구사항 명세 3개 작성
- 5.3 정형적인 형식에 따른 유스케이스 작성

6장. 시스템 설계

- 6.1 설계 개념의 중요한 개념을 적용
- 6.2 설계 모델에 따른 요소별 설계

7장. 아키텍처 개념

- 7.1 아키텍처 스타일 선정 및 이유
- 7.2 아키텍처 설계 프로세스 정의 및 설계

8장. 품질

- 8.1 시스템 품질 속성 정의
- 8.2 비즈니스 품질 속성 정의
- 8.3 아키텍처 품질 속성 정의
- 8.4 소프트웨어 품질 목표, 속성과 척도
- 8.4 통계적 방법 이용 사례 선정

9장 . 테스트 전략

- 9.1 테스트 전략적 이슈 순위정의 (전략 성공)
- 9.2 통합테스트 방법 및 순서
- 9.3 시스템 테스트 중요이슈 정의
- 9.4 소프트웨어결함 유형 정의

Chapter 10. 프로젝트 관리

10.1 우리 팀의 융합적인 요소

10.2 우리 팀의 독소 요소

10.3 프로젝트 스케줄링의 활동 네트워크 작성

1) 타스크 일정 및 의존성

2) 활동 네트워크