

소프트웨어공학

소프트웨어 프로젝트 관리

1 소프트웨어 프로젝트 관리 개념

2 프로젝트 관리 대상

3 메트릭스

4 프로젝트 관리의 구성 단계

소프트웨어 프로젝트 관리 개념

1. 프로젝트의 정의

- A project is a temporary endeavor undertaken to create a unique product, service, or result.
- 프로젝트는 고유한 제품, 서비스, 결과물을 만들기위해 수행되는 일시적인 노력이다.
- “Temporary” → 모든 프로젝트가 정해진 기간을 갖고 있음을 의미한다.
- “Unique” → 프로젝트를 통해 창출된 제품 또는 서비스가 다른 유사한 제품 또는 서비스와는 구분

소프트웨어 프로젝트 관리 개념

1. 프로젝트의 정의

- **프로젝트가 포함되는 것들의 예**
 - 새로운 제품 또는 서비스의 개발
 - 구조, 스텝 또는 조직의 스타일 변경에 영향
 - 새로운 운송 수단의 설계
 - 새로운 것 또는 변경된 정보 시스템의 개발 또는 획득
 - 건물 또는 설비의 건설
 - 정부 기관에 대한 캠페인 수행
 - 새로운 업무 절차 또는 프로세스의 구현

소프트웨어 프로젝트 관리 개념

1. 프로젝트의 정의

- ‘Temporary’
 - 모든 프로젝트는 확정된 시작일과 종료일을 갖는 것을 의미
 - 프로젝트 기간이 짧아야 하는 것을 의미하지는 않음
 - 제품이나 서비스에만 적용되는 것은 아님
- ‘Unique’ 생산품 및 서비스
 - 프로젝트의 생산품은 유일

소프트웨어 프로젝트 관리 개념

2. 프로젝트 관리(Project Management)의 정의

- Project Management is the application of knowledge, skills, tools, and techniques to project activities to meet the project requirements.
- 프로젝트 관리는 고객의 요구사항을 완수하고자 프로젝트 활동에 지식, 기술, 도구 및 기법 등을 응용하여 프로젝트 절차에 적용하는 일련의 관리 방법이다.
- 소프트웨어 프로젝트를 조직하고(organizing) 계획하고(planning) 일정관리 (scheduling)를 하는 것이다.
- 주어진 기간 내에 최소의 비용으로 사용자를 만족시키는 시스템을 개발하기 위한 전반적인 활동

소프트웨어 프로젝트 관리 개념

2. 프로젝트 관리(Project Management)의 정의

- **소프트웨어 개발 계획을 세우고 분석, 설계, 구현 등의 작업을 통제하는 것**
→ 소프트웨어 생명주기의 전 과정 걸쳐 진행
- **소프트웨어 프로젝트를 성공적으로 수행**
 - 수행할 작업의 범위, 필요한 자원, 수행업무, 이정표, 비용, 추진 일정 숙지

프로젝트 관리 대상

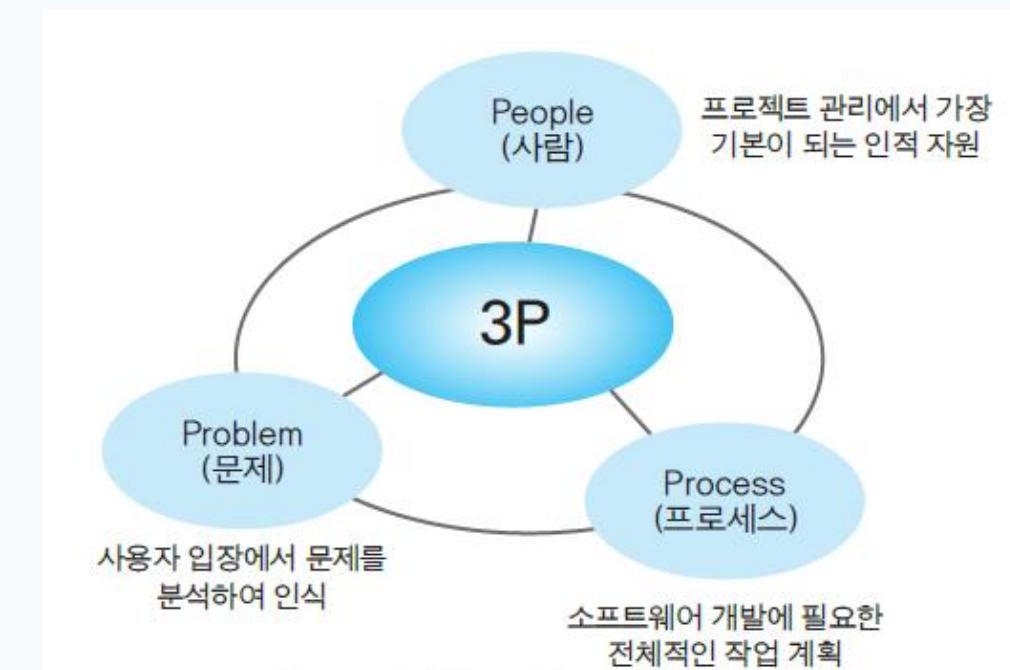
1. 프로젝트 관리는 다음과 같은 관리를 유기적으로 결합하여 구성

- 계획 관리 : 프로젝트 계획, 비용 산정, 일정 계획, 조직 계획
- 품질 관리 : 품질 통제, 품질보증
- 위험 관리 : 위험 식별, 위험 분석 및 평가, 위험 관리 계획, 위험 감시 및 조치관리의 대상

프로젝트 관리 대상

2. 효과적인 프로젝트 관리를 위한 3P(3대 요소)

- 사람(People)
 - 프로젝트 관리에서 가장 기본이 되는 인적자원
- 문제(Problem)
 - 사용자 입장에서 문제를 분석하여 인식함
- 프로세스(Process)
 - 소프트웨어 개발에 필요한 전체적인 작업 계획



프로젝트 관리 대상

2. 효과적인 프로젝트 관리를 위한 3P(3대 요소)

- 사람(People)
 - 인적 자원 관리 성숙도 모델(PM-CMM)
 - ✓ 복잡한 어플리케이션을 수행하는 데 조직의 구성을 쉽게 파악하도록 하기 위함
 - ✓ 사원 채용, 선택, 능력 관리, 교육, 보상, 경력 개발, 조직 및 작업 설계, 팀 구성
 - 프로젝트 수행자
 - ✓ 선임 관리자, 프로젝트 관리자, 실무자, 고객(의뢰인), 일반 사용자

2. 효과적인 프로젝트 관리를 위한 3P(3대 요소)

- 문제(Problem)
 - 목적
 - ✓ 프로젝트 목적 및 범위 정의
 - ✓ 프로젝트의 전체 목표 식별
 - 범위
 - ✓ 문제의 특성을 설명하는 전후 문맥, 목적, 기능 및 성능
 - ❖ 소프트웨어를 어떻게 구축하고 어떤 제약 사항이 존재하는가?
 - ❖ 고객이 요구하는 정보는 어떠한 소프트웨어의 출력으로 생성되는가?
 - ❖ 입력 데이터가 어떤 기능을 통해 출력데이터로 변환되는가?
 - ❖ 어떠한 성능이 요구되는가?

2. 효과적인 프로젝트 관리를 위한 3P(3대 요소)

- 문제(Problem)
 - 문제 분할
 - ✓ 관리를 용이하도록 하기 위해 복잡한 문제를 작은 문제들로 분할
 - ❖ 기능 분할
 - ❖ 프로세스 분할

프로젝트 관리 대상

2. 효과적인 프로젝트 관리를 위한 3P(3대 요소)

- **프로세스(Process)**
 - 태스크들의 집합
 - ✓ 태스크, 일정표, 산출물 및 품질 보증
 - 프로세스 분할
 - ✓ 프로젝트에 적합한 프로세스를 선택
 - ✓ 문제와 프로세스를 조합

1. 메트릭스(Metrics)

- IEEE 정의
 - 시스템, 컴포넌트 또는 프로세스가 주어진 속성을 어느 정도 만족하는가에 대한 수량적인 측정
- Measure
 - 프로젝트나 프로세스에서 측정된 어떤 속성의 정도, 양, 치수(크기)
- Measurement(측정)
 - Measure를 결정하는 행위
- Metrics (메트릭스)
 - 소프트웨어 측정을 통하여 소프트웨어의 품질, 생산성 또는 개발 비용 등을 추정할 수 있는 척도

2. 소프트웨어 측정

- **소프트웨어를 측정하는 이유**

- 소프트웨어의 품질을 제시하기 위해
- 새로운 개발 방법이나 도구를 사용하여 얻는 이점을 측정하기 위해
- 추정을 위한 기준선을 결정하기 위해
- 새로운 도구나 추가적인 교육이 필요한 지 결정하기 위해

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 소프트웨어나 프로세스에 대한 직접적인 측정

프로젝트	LOC	노력 (월인원)	비용 (천만원)	문서의 페이지	오류 (Errors)	결함 (Defects)	인원
A	13,400	20	8	373	128	30	2
B	29,100	63	48	1317	334	92	7
C	19,500	30	35	960	274	70	5
...

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- **측도**

- 오류 발생율(Error Rate): KLOC 당 오류의 수
- 결함 발생율(Defect Rate): KLOC 당 결함의 수
- 비용 효율성(Cost Effectiveness) : LOC 당 비용(원)
- 문서화(Documentation) : KLOC 당 문서의 페이지 수
- 기타

- ✓ 오류의 수/월 인원
- ✓ LOC / 월 인원
- ✓ 원 / 문서의 페이지 수

장점	단점
<ul style="list-style-type: none">• LOC 중심으로 측정• LOC는 쉽게 계산이 가능• 많은 측정 모델이 LOC를 중요한 입력값으로 사용	<ul style="list-style-type: none">• 프로그래밍 언어에 따라 크기가 가변적• LOC의 기준이 모호하고 표준이 결여

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 기능 중심 메트릭스(Function-Oriented Metrics)
 - 기능을 중심으로 측정
 - ✓ 다른 측정을 사용해서 간접적으로 유도
 - Function Point (FP)
 - ✓ 1979년 Albrecht가 제안
 - ✓ 소프트웨어의 직접 측정을 바탕으로 과거의 프로젝트 경험 자료를 이용하여 유도

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 기능 중심 메트릭스(Function-Oriented Metrics)
 - 5가지 측정 항목
 - ✓ 사용자 입력 개수(Number of User Inputs)
 - ✓ 사용자 출력 개수(Number of User Outputs)
 - ✓ 사용자 질의 개수(Number of User Inquiries)
 - ✓ 파일 개수(Number of Files)
 - ✓ 외부 인터페이스 개수(Number of External Interfaces)
 - ❖ 기계가 판독 가능한 인터페이스의 수

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 기능 중심 메트릭스(Function-Oriented Metrics)

- 측정 단계

- ✓ 단계 1. Function Point 테이블

- ❖ 각 매개변수의 개수, 복잡도, 점수

- ✓ 단계 2. 복잡도 조정값

- ❖ 인자에 비율 부여 : 0 (No Influence) to 5 (Significant)

- ✓ 단계 3. Function Point 계산

- ❖ 매개변수별 복잡도와 가중치를 이용하여 계산

매개변수	Count	단순	평균	복잡	= []
사용자 입력	[]x	3	4	6	= []
사용자 출력	[]x	4	5	7	= []
사용자 질의	[]x	3	4	6	= []
파일수	[]x	7	10	15	= []
외부 인터페이스	[]x	5	7	10	= []

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 기능 중심 메트릭스(Function-Oriented Metrics)

- 장단점

장점	단점
<ul style="list-style-type: none">프로그래밍 언어에 독립적	<ul style="list-style-type: none">계산이 주관적인 자료를 바탕으로 함물리적인 의미가 없음

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- 메트릭스 조정
 - LOC와 FP간의 관계
 - ✓ 프로그래밍 언어에 따라 다르다.

프로그래밍 언어	LOC/FP
Assembly Language	320
C	128
Cobol	105
Fortran	105
Pascal	90
Ada	70
Object-Oriented Languages	30
4GLs	20
Code Generators	15
Spreadsheets	6
Graphical Languages (icons)	4

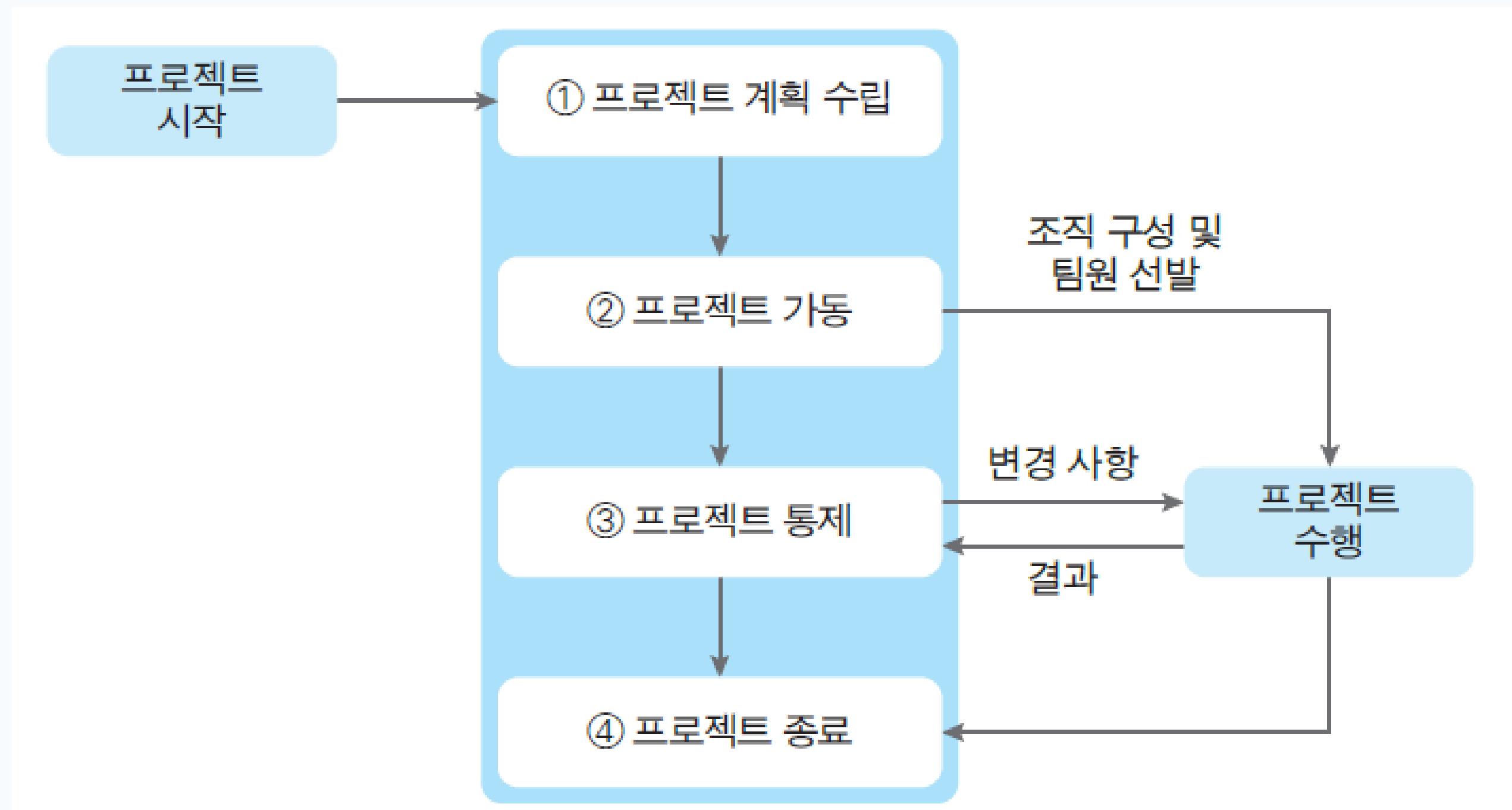
- ✓ LOC와 FP는 생산성 메트릭을 유도하는데 사용됨

3. 크기 중심 메트릭스(Size-Oriented Metrics)

- **메트릭스 조정**
 - LOC와 FP간의 관계
 - ✓ 소프트웨어 생산성에 영향을 주는 주요 인자
 - ❖ 인적 인자(People Factors)
 - ❖ 문제 인자(Problem Factors)
 - ❖ 프로세스 인자(Process Factors)
 - ❖ 제품 인자(Product Factors)
 - ❖ 자원 인자(Resource Factors)

04

프로젝트 관리의 구성 단계



프로젝트 관리의 구성 단계

1. 프로젝트 계획 수립

- 프로젝트의 목적을 기술하고, 이를 달성하기 위해 필요한 업무와 성취해야 할 일들을 결정한다.
- 프로젝트를 정의하고 프로젝트 일정 계획, 소요 자원 예측, 위험 평가, 프로젝트에 대한 승인을 얻어낸다.

2. 프로젝트 가동

- 프로젝트가 수행될 환경을 구성하고 프로젝트에 참여할 인력을 교육시킨다.
- 프로젝트를 진행할 조직을 구성하고 각 팀원을 선발한다.

프로젝트 관리의 구성 단계

3. 프로젝트 통제

- 계획 대비 프로젝트의 척도(metric)를 점검하고, 변경 사항을 승인하는 등의 작업을 수행한다.
- 프로젝트 전 기간 동안 수행된다.

4. 프로젝트 종료

- 수행 결과의 완전성을 점검하고 프로젝트를 종료한다.