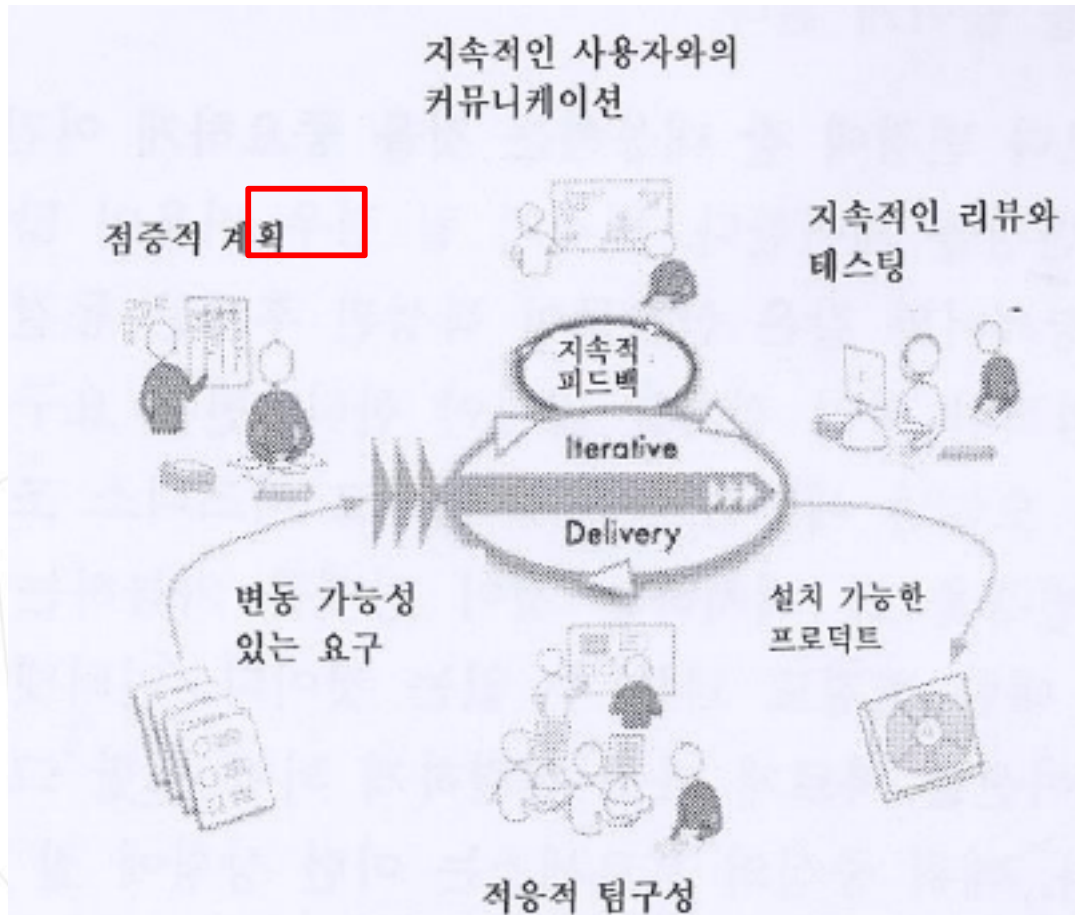


- 폭포수 프로세스의 단점을 해결
- 절차와 도구보다 **개인(팀원들간)과 소통을** 중요시 한다
- 잘 쓴 문서보다는 실행되는 **소프트웨어에 더 가치를** 둔다
→ 품질 보다는 일정
- 계약 절충보다는 **고객(사용자) 협력을** 더 중요하게 여긴다
- 계획을 따라 하는 것보다 **변경에 잘 대응하는 것을 중요하게** 여긴다

애자일 프로세스

- 사용사례 또는 사용자 스토리 이나 피처(feature) 단위
- 테스트 중심 개발(TDD: Test Driven Development): 모든 기능에는 **유닛 테스트 코드가 먼저 작성되어야 함**

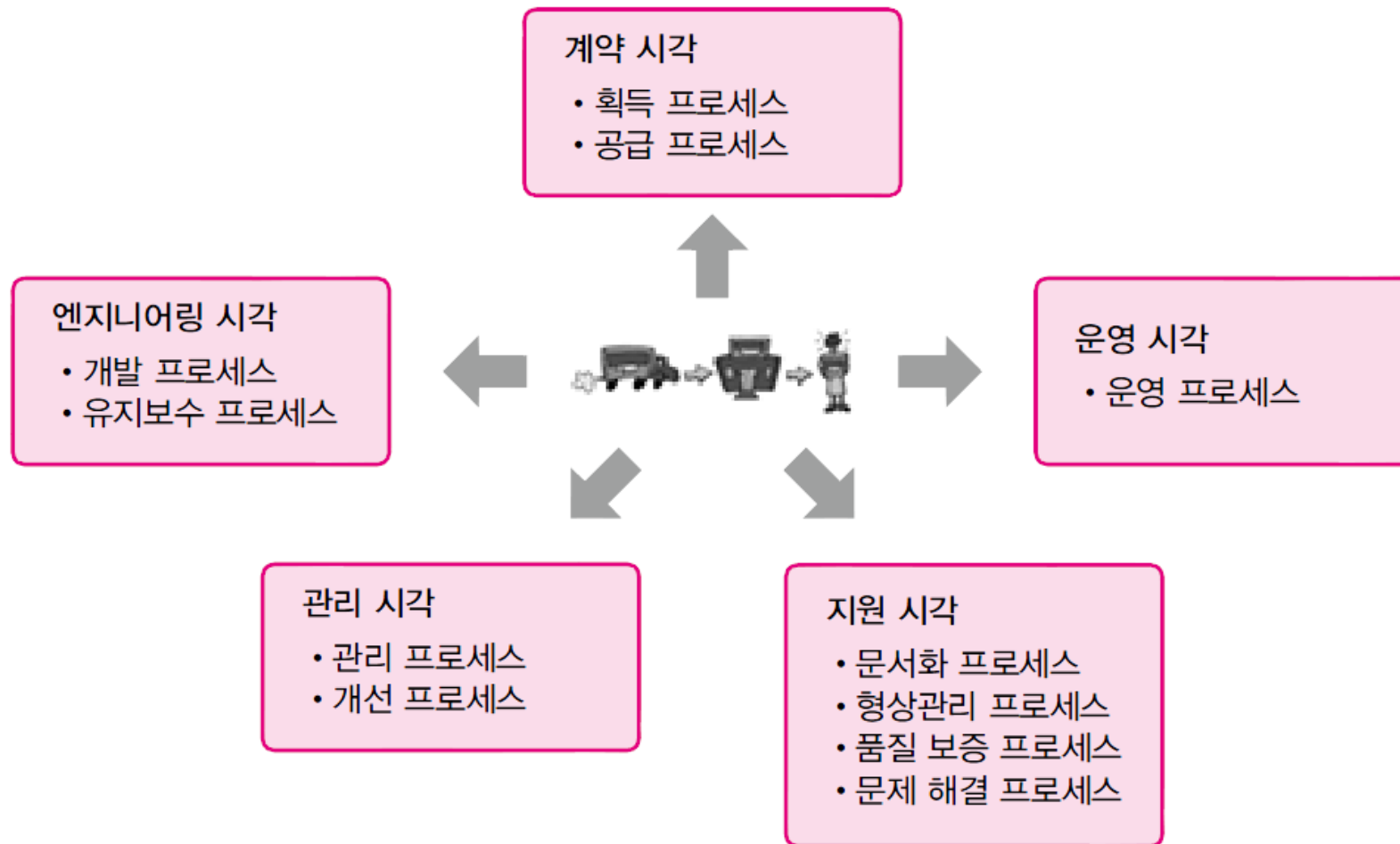


Feature: A small, discrete piece of functionality desired by stakeholders (or clients).

For example, "Calculate the total of a sale", "Validate the password of a user", "Authorize the sales transaction of a customer".

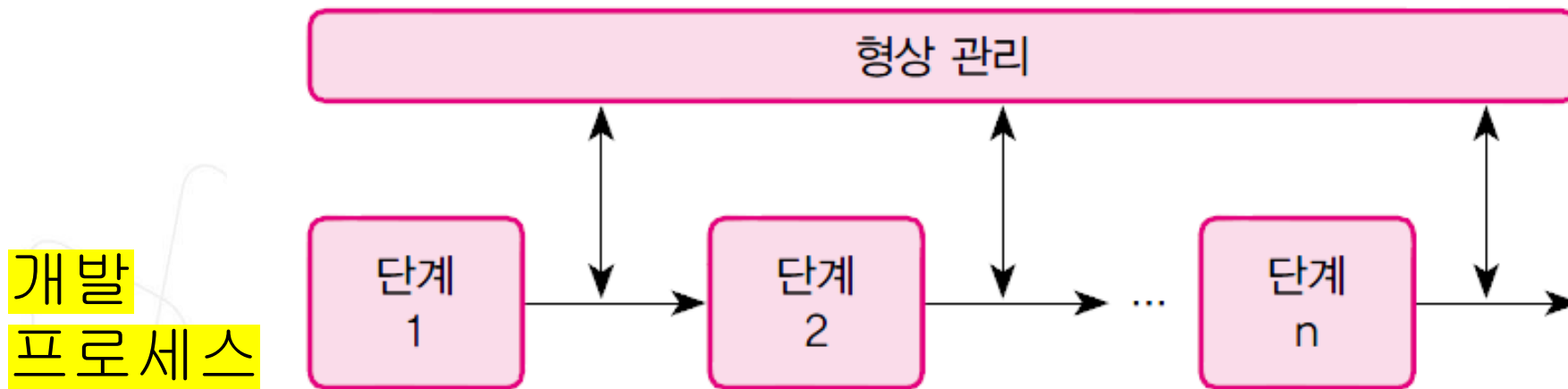
2.4 지원 프로세스

- ISO/IEC 12207에서의 프로세스 그룹



형상 관리 프로세스

- 개발 중에 발생하는 **변경**을 체계적으로 컨트롤 하는 것
- 개발작업과 독립적인 작업



(1) 형상 관리 기능

- 프로그램의 최신 버전 유지 (여러 버전들 중 누가 최신인가?)
- 지정된 버전으로 되돌아 갈 수 있는 기능
- 무허가 (unapproved) 변경이나 삭제를 방지
- 현 시스템에 대한 모든 정보, 문서 등의 정보를 모아 보관

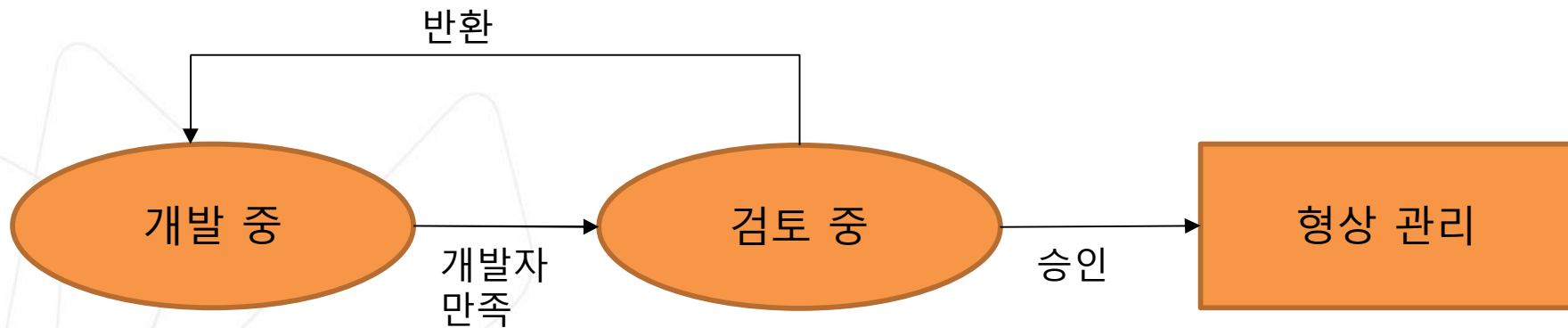
(2) 형상관리 메커니즘

- 프로젝트에서 변경이 발생되었을 때 처리하는 시나리오를 다루는 메커니즘을 제공

- 형상 관리 대상 파악과 베이스라인 지정
- 버전관리
- 접근제어

- 형상 관리 아이템의 생명 주기

A **baseline** is a static (i.e. unchanging) **snapshot** of a set of predefined work products (formally **approved**) at any point in time.



2.5 방법론

- 방법론

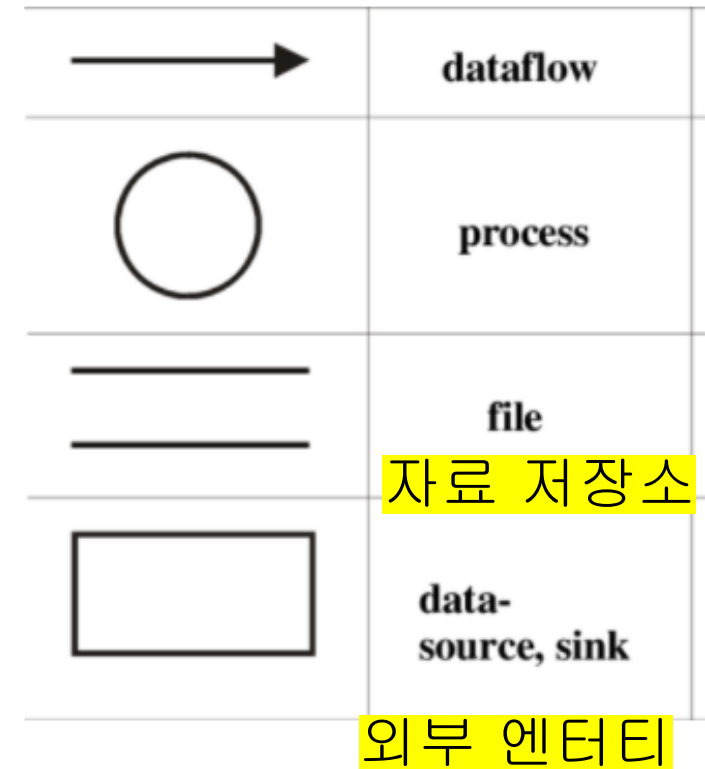
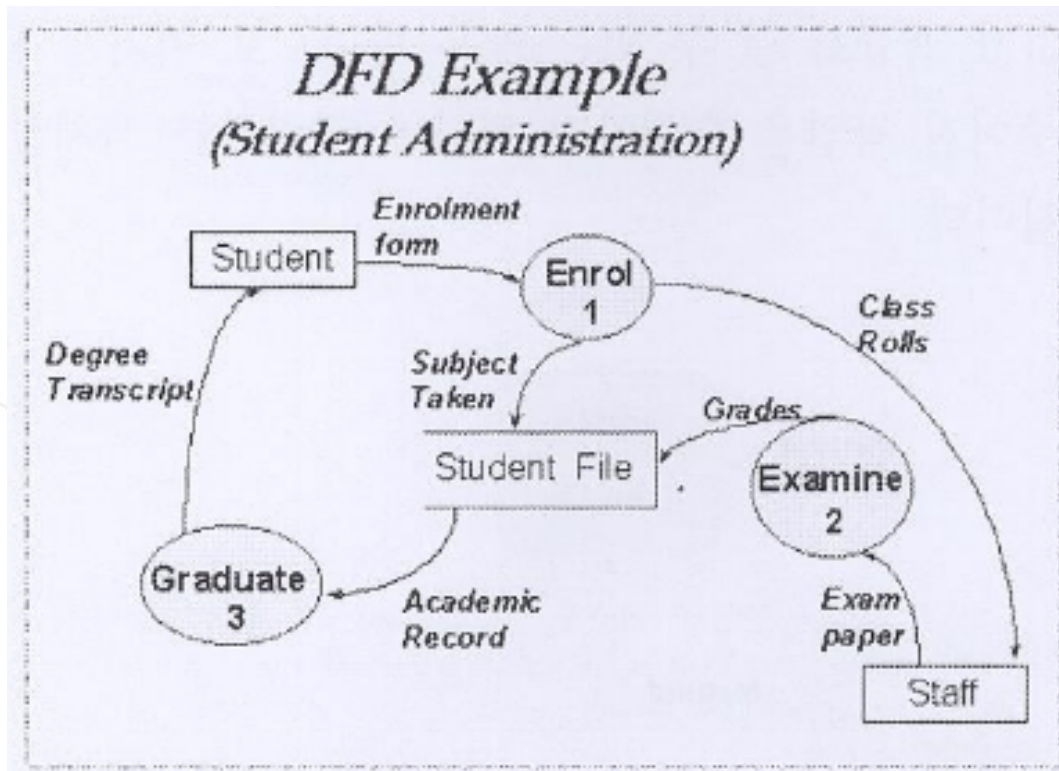
- 소프트웨어 프로세스의 각 작업을 어떻게 수행하느냐를 정의

- 프로세스

- 일반적으로 개발할 때 하여하는 할 작업만을 명시
- 어떤 관계가 있는지 나타내지 않음

구조적 방법론

- 분리와 정복(divide and conquer) 원리 적용
- 자료 흐름도(**DFD: Data Flow Diagram**)를 구조도 (Structure Chart)로 변경하는 과정
 - 구조도 : 모듈 사이의 관계를 나타내는 그래프



객체지향 방법론

- 자료와 함수를 가까운 곳에 정의하여 객체(object)로 묶어두고(encapsulation) 객체 사이에 메시지를 호출 전달하여 (메소드를 호출하여) 원하는 기능을 담당하게 하는 것
- 객체지향 패러다임

