

플랫폼 개발 생명주기 (PDLC: Platform Development Life Cycle)

❖ Terminology (1)

✓ 시스템(**System**)

- 시스템은 하나의 비즈니스 기능에 사용되는 상호 연관된 절차들의 그룹으로 구별 가능한 경계를 가짐. 즉, 통합 된 전체를 형성하는 규칙적으로 상호 작용하거나 상호 의존적인 단위 그룹입니다. 모든 시스템은 공간과 시간적 경계에 의해 묘사되고, 환경에 의해 영향을 받으며, 그 구조와 목적에 의해 기술되고 그 기능으로 표현된다
- 정보를 처리하는 데 이용되는 프로그램들, 절차들, 데이터 및 기기들의 다양한 집합이나 조합을 가리키기도 함. (예로, 회계 관리 시스템, 요금 관리 시스템, 온라인 데이터베이스 관리 시스템)

✓ 서브시스템(**SubSystem**)

- 서브 시스템은 시스템 자체이자 더 큰 시스템의 구성인 요소 집합입니다. 서브 시스템은 시스템이 제어하는 운영 환경의 특성을 정의하는 정보를 포함하는 시스템 오브젝트입니다

❖ Terminology (2)

✓ Component (컴포넌트)

- 독립적인 단위/소프트웨어 모듈. 시스템은 모듈로 구성된 컴포넌트로 나뉨.
- 특정 기능이나 관련된 기능들의 조합. 예로, GUI에서 하나의 단추, 이자계산용 소형 계산기 등

✓ Function (함수)

- 소프트웨어에서 특정 동작을 수행하는 일정 코드 부분을 의미 (기능, 역할)
- 하나의 특별한 목적의 작업을 수행하기 위해 독립적으로 설계된 **코드(명령어)**의 집합
- 소프트웨어에서 함수를 사용하는 가장 큰 이유는 반복적인 프로그래밍을 피할 수 있기 때문.

❖ Terminology (3)

✓ Module (모듈)

- **독립적인 하나의 소프트웨어 혹은 하드웨어 요소.** 프로그램의 기능을 독립적인 부품으로 분리한 것을 모듈이라고 하며, 모듈은 일반적으로 서브루틴과 데이터 구조의 집합체로서, 그 자체로서 컴파일 가능한 단위이며, 재사용 가능.
- 하나의 시스템에서 각 모듈은 유일한 이름을 가지며 모듈에서 또 다른 모듈을 호출할 수 있음. 즉, 모듈들은 하나 이상의 모듈과 상호연관관계를 가짐
- 모듈은 하나 이상의 기능을 가질 수 있음. 예로, 1개의 기능을 가지는 예시로는 저장, 분석, 처리 등 기능 각각 가지며, 2개 이상의 기능을 가지는 예시로는 송/수신, 생성/삭제, 압축/해제 등이 있음
- **모듈은 하나 이상의 루틴을 포함하는 소프트웨어 구성 요소 또는 프로그램의 일부.** 독립적으로 개발된 하나 이상의 모듈이 프로그램을 구성. 엔터프라이즈급 소프트웨어 응용 프로그램에는 여러 가지 다른 모듈이 포함될 수 있으며 각 모듈은 고유 한 별도의 비즈니스 작업을 수행함. 모듈을 사용하면 프로그래머가 소프트웨어 응용 프로그램 기능의 한 영역에만 집중할 수 있으므로 프로그래머의 작업이 쉬워짐. 모듈은 일반적으로 인터페이스를 통해 프로그램(소프트웨어)에 통합됨.

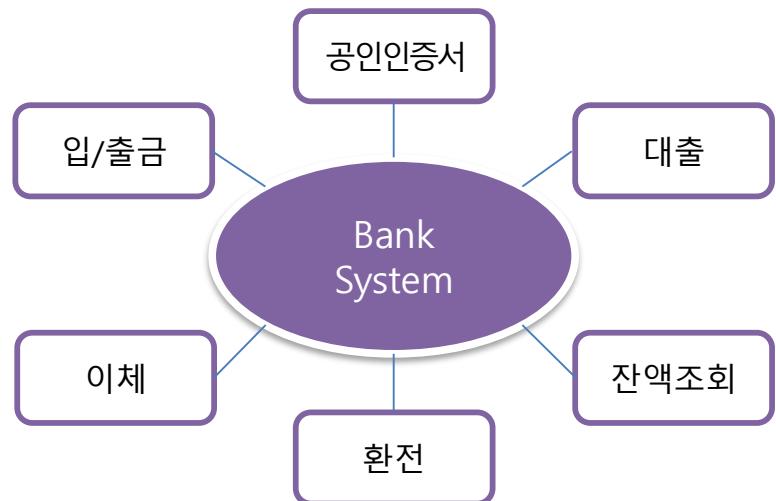
❖ Terminology (4)

✓ Modularization (모듈화)

- The action or process of making something modular; construction on modular principles.
- 생산성과 최적화, 관리에 용이하게 모듈(기능) 단위로 분할하는 것. 개념적으로 정의된 시스템이나 프로그램을 기능별로 분해하여 각각의 모듈로 정의하는 것이며, 하나의 시스템에서 정의된 모듈은 중복되지 않아야 하고 각 모듈은 하나 이상의 모듈과 연관관계를 가지고 있어야 함
(예1: 자동차 개발) 차량의 운전 기능, 개폐 기능, 조명 기능, 제어 기능 등
(예2: 은행 시스템) 은행 시스템의 입/출금, 이체, 환전, 잔액조회 등 기능 등



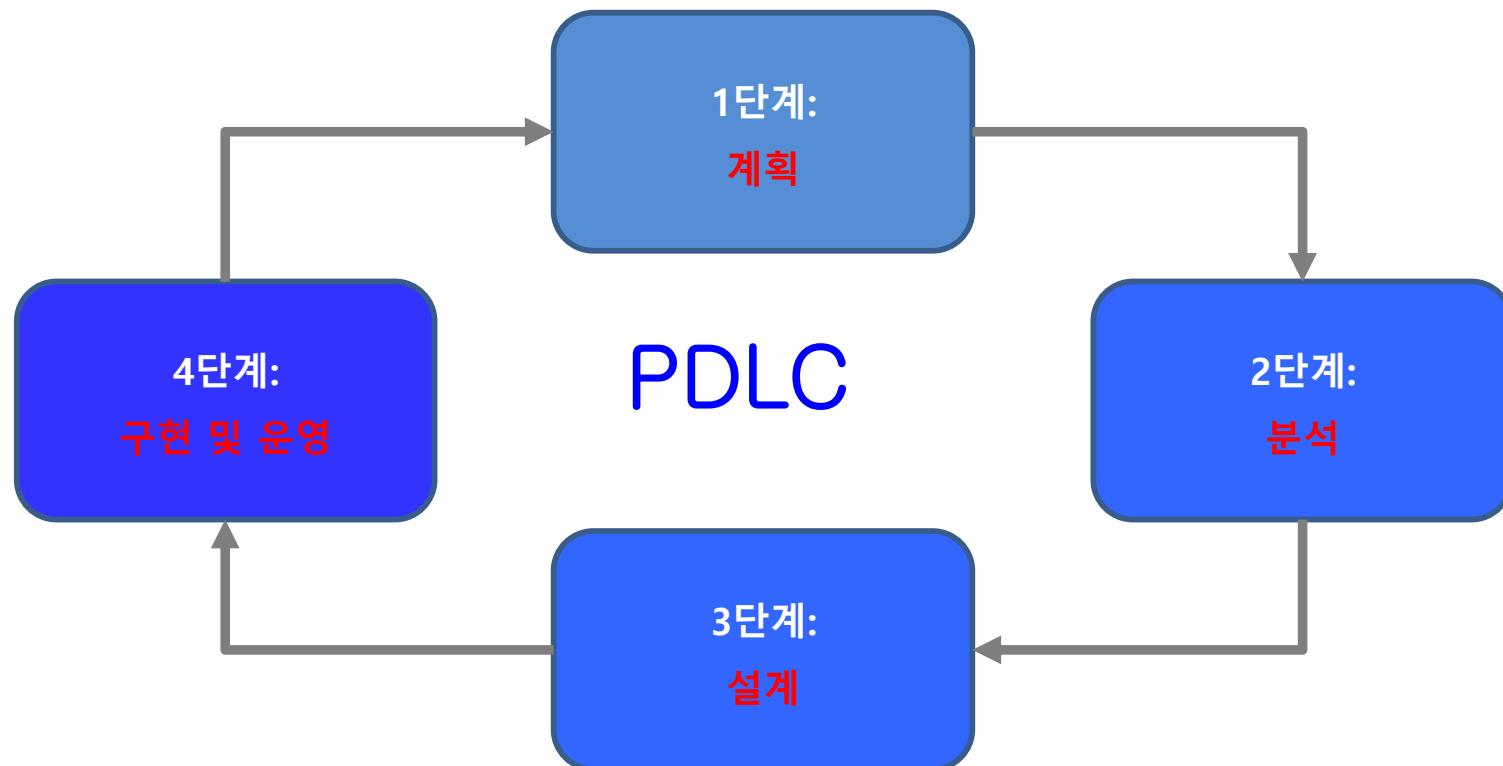
<자동차 개발에서의 모듈화>



<은행 시스템 개발에서의 모듈화>

❖ PDLC(Platform Development Life Cycle)

- ✓ 계획, 분석, 설계, 구현 4단계로 이루어진 SDLC의 기본 틀을 기반으로 플랫폼의 목적, 환경, 구성에 맞게 설계하고 개발할 수 있는 '플랫폼' 개발 생명주기



1 단계 : 계획 (1/3)

- ✓ 플랫폼에 대한 모든 니즈를 분석하고 정렬하며, 플랫폼에 대한 잠재적 프로젝트를 정의하고, 그 플랫폼의 지속 여부를 결정
- ✓ **플랫폼 개발 계획 단계의 개발 프로세스**

① 플랫폼의 확인과 선정

- 개발자는 3개의 주요 요구에서 플랫폼 주제를 선정할 수 있음
 - I) 새로운 서비스를 제공할 플랫폼의 개발
 - 2) 기존의 운영중인 플랫폼의 개선
 - 3) 기존의 운영중인 플랫폼보다 더 나은 운영 환경의 개발
- 선택한 주제에 관련한 시장 조사 및 경쟁 업체 등의 벤치마킹을 실시

② 플랫폼의 초기화와 계획

- 플랫폼의 전체적인 방향 및 계획에 대한 수립
- **플랫폼 개발 팀의 결성, 플랫폼 개발 환경 결정, 예비 예산 분석, 사용자와의 관계 설정, 기본 개발 계획의 수립 등**

1 단계 : 계획 (2/3)

③ 실현 가능성(feasibility/realizability) 평가

- 플랫폼 개발에 있어서 필요한 요소를 고려하여 실현 가능성을 평가

경제적	운영적	기술적	시간적	법적, 계약적	정략적
개발 비용 및 예산	개발자 인원 및 역량	기술 보유 현황, 실현 가능성	비용 및 인력에 맞는 프로젝트 일정	특허 침해 및 위법 여부	프로젝트 핵심 인력의 자체평가

④ 기본 플랫폼 개발 계획 구축

- 앞서 계획한 초기화 계획 및 실현 가능성에 대한 분석을 통해 전체적인 계획을 수립
- 개발하고자 하는 플랫폼의 설명, 실현 가능성 분석, 개발 일정 계획 등을 작성

⑤ 기본 플랫폼 개발 계획의 검토

- 내부 회의를 통해 앞서 계획한 모든 사항들에 대한 검토와 보완을 실시

1 단계 : 계획 (3/3)

✓ 두 가지 주요 활동

① 보다 새롭거나 발전된 플랫폼의 필요성을 확인

- 제안하는 플랫폼이 기존 플랫폼보다 성능이 좋은가? – 속도, 정확도 등
- 사용자의 불편을 해소할 수 있는가?
- 현 시점에서 어디에서도 제공되지 않는 기능을 포함하고 있는가?

② 플랫폼을 조사하고 제안된 플랫폼의 범위를 결정

- 유사한 플랫폼이 존재하는가?
- 기능은 어디까지 제공 할 것인가? - 개발자의 능력 및 자본을 고려
(예) 모니터 개발 – 해상도, 스피커기능, 흑백기능, 절전모드 등

※ 주요 산출물 : 플랫폼 프로젝트 제안서 – 배경, 목표, 계획, 범위, 유사 사례 등