

## 5주차 3차시 소프트웨어 개발 방법론과 응용 소프트웨어

### 【학습목표】

1. 소프트웨어 개발 방법 절차와 종류를 살펴보고 설명할 수 있다.
2. 응용 소프트웨어의 종류와 기능을 살펴보고 설명할 수 있다.

### 학습내용1 : 소프트웨어 개발 방법론

\* 모델하우스에 가보면 아파트의 구조와 재료가 서로 다른 모델이 있는 것과 같이 소프트웨어 개발 방법도 여러 가지가 있다는 점을 강조

#### 1. 소프트웨어 공학

1) 일반적으로 공학 분야는 이미 존재하는 부품("off-the-shelf" component)를 가져와서 제작, 그러나 이러한 방법이 소프트웨어에는 잘 적용되지 않음(커스터 마이징이 필요)

2) 소프트웨어는 개발보다 유지보수에 많은 비용

① CASE(Computer-Aided Software Engineering) 도구

- 프로젝트 기획, 관리, 프로토타입 개발 및 시뮬레이션, 인터페이스 설계 및 프로그램 설계에 지원

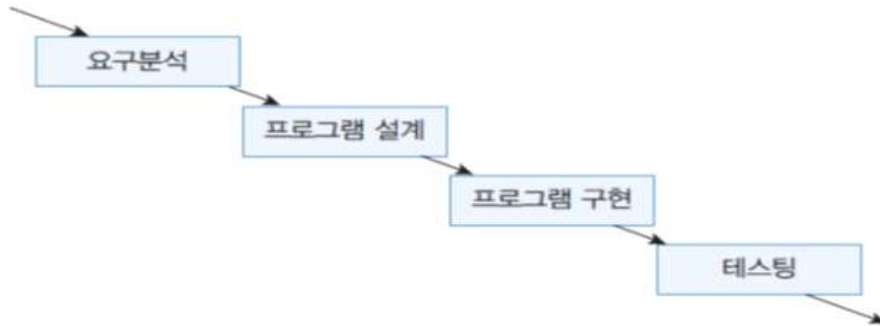
② 통합개발환경(IDE: Integrated Development Environment)

- 에디터, 컴파일러, 디버깅 도구 등의 개발 도구가 통합된 형태로 제공

## 2. 소프트웨어 개발 방법론

### 1) 소프트웨어 라이프 사이클(Software Life Cycle)

#### ① 폭포수 모델(Waterfall Model)



[전통적인 소프트웨어 라이프사이클]

### 2) 소프트웨어 공학 방법론

#### ① 폭포수 모델

- 요구분석, 프로그램 설계, 프로그램의 구현, 테스트 과정

#### ② 점진적 모델(Incremental Model)

- 처음에는 간단한 시스템, 점진적으로 사용자들로부터 검증을 거친 후 현실에 맞게 새로운 기능 추가 및 증가

#### ③ 오픈소스 모델

- 무료 공개 소프트웨어 개발 방식
- 예) Linux는 초기 토발즈에 의해 커널 부분만 개발, 공개 후 많은 프로그래머들이 다양한 기능 추가로 버전 업

#### ④ 애자일 방법론(Agile Method)

- 시시각각으로 변화하는 비즈니스 환경과 상황을 소프트웨어 개발에 기민하게 반영
- 익스트림 프로그래밍(EP: Extreme Programming): 10여명 이하의 개발팀을 구성하여 서로 아이디어와 디자인을 상호 공유할 수 있는 환경
  - 단시일 내에 SW를 개발한 후 매일 새로운 요구와 상황을 만족시키는 방향으로 발전
  - 소프트웨어의 모듈화 : 프로시저, 클래스, 컴퍼넌트

### 3. 소프트웨어 개발 도구

1) 소프트웨어 개발 사이클의 전 과정을 효율적으로 진행하고 개발 과정을 최대한 자동화

#### ① CASE 도구



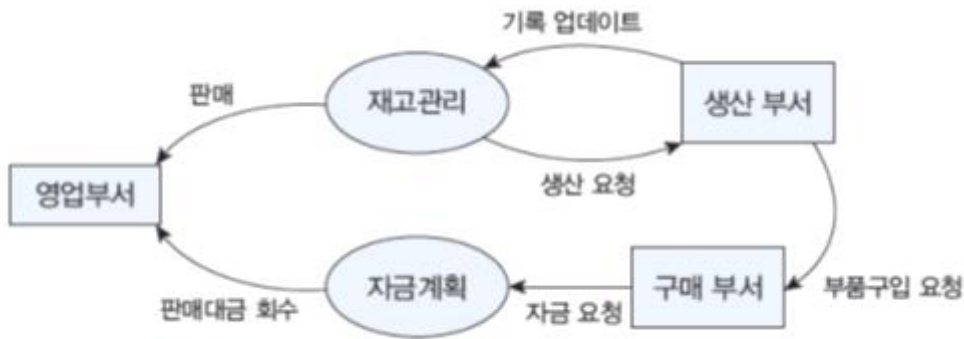
[프로그램개발 라이프사이클(PDLC)과 CASE 도구]

- 절차적 프로그래밍과 객체지향 프로그래밍에 따라 다르게 적용

#### ② 데이터흐름 다이어그램, 데이터 사전

- 데이터흐름 다이어그램(DFD; Data Flow Diagram)

- 주로 절차적 프로그램의 개발
- 데이터가 어떤 방향으로 흘러가는지, 데이터 발생 지점이 어디인지에 초점



[간단한 데이터흐름 다이어그램의 예]

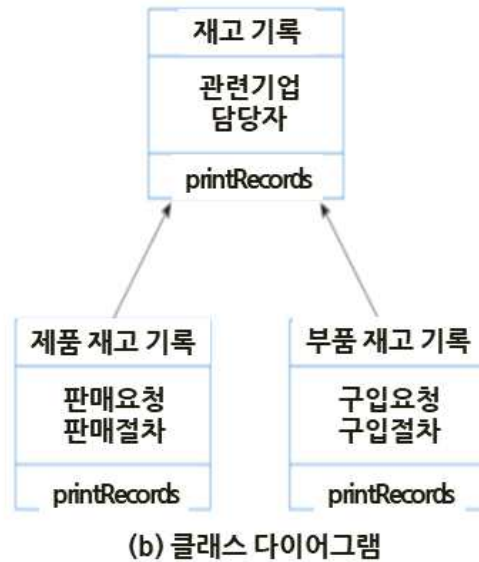
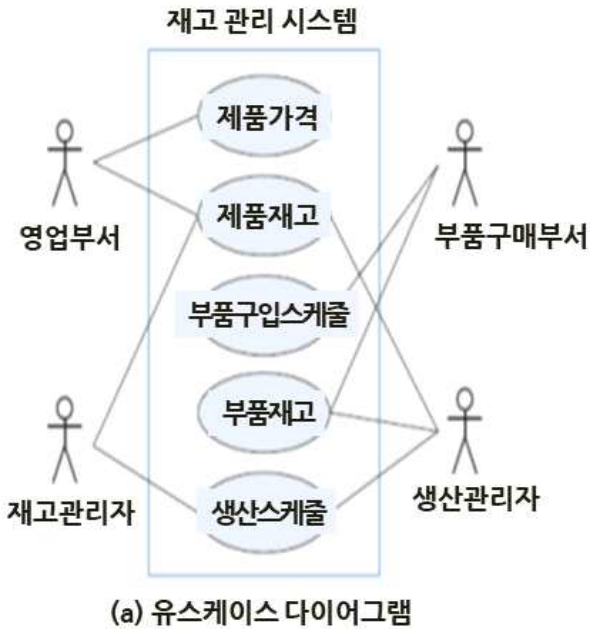
- 데이터 사전(Data Dictionary)

- 소프트웨어의 식별자(Identifier)가 어디에 정의되어 있고 어떤 모듈에서 사용되며 데이터 아이템이 어디에 저장되어 있는지

#### ③ 통합 모델링 언어(UML: Unified Modeling Language)

- 객체지향 소프트웨어 개발에서 주로 사용
- 목적에 따라 사용되는 많은 그래픽 다이어그램
  - 사용자 관점(User View)
  - 구조적 관점(Structure View)
  - 행위적 관점(Behavioral View)
  - 구현 관점(Implementation View)

- UML 다이어그램의 사례
  - 사용자 관점: 유스케이스 다이어그램
  - 구조적 관점: 클래스 다이어그램



#### 4. 소프트웨어 테스트

##### 1) 소프트웨어의 품질 보장(Quality Assurance)

###### ① 유리박스 테스트

- 가장 중요하고 문제의 소지가 있는 모듈을 중심으로 테스트(내부 구조를 잘 이해할 때 사용)
- ‘파레토 법칙(Pareto’s Rule)’ : ”80~20 법칙“
- 기본경로 테스트 : 모든 가능한 실행루트를 적어도 한번 이상 실행해서 검증

###### ② 블랙박스 테스트

- 소프트웨어의 내부를 이해하지 못해도 가능한 모든 데이터 세트를 사용하여 소프트웨어의 작동을 검증
- 경계 값 분석(Boundary Value Analysis)
- 내부 개발팀 중심의 알파 테스트(Alpha Testing) 및 외부 사용자중심의 베타 테스트(Beta Testing)

## 학습내용2 : 응용 소프트웨어

- 스마트폰의 바탕화면의 앱도 대부분 응용 소프트웨어임을 강조

### 1. 응용 소프트웨어의 활용 분야와 특징

#### 1) 응용 소프트웨어의 활용 분야



#### 2) 응용 소프트웨어의 배포 방식

##### ① 번들 소프트웨어

- PC 출현 이전인 1970년대까지, 운영체제나 문서 작성기 등이 내장되어 하드웨어와 함께 제공

##### ② 패키지 소프트웨어

- PC 출현 이후 패키지로 포장하여 판매(오피스 프로그램)

##### ③ 웹 다운로드 방식

- 웹 다운로드 소프트웨어 배포 방식, 업그레이드 버전에 유리

- 최근에는 임대해서 비용을 줄임 : SaaS(Software as a Service)

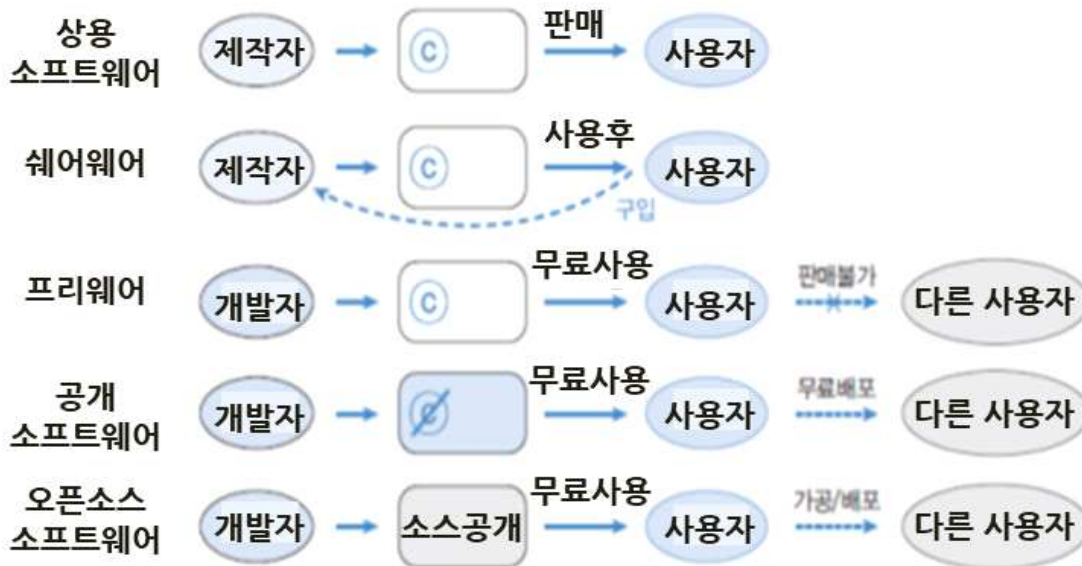
##### ④ 맞춤형 방식

- ERP, SCM, MIS와 같은 정보시스템의 경우 기본 기능을 포함하는 공통부분을 미리 개발하고 고객에 알맞은 부분을 추가하여 제공

### 3) 응용 소프트웨어의 소유권/저작권에 따른 분류

- ① 상용 소프트웨어
- ② 셰어웨어(Shareware) : 일정기간 사용해보고 구매하는 소프트웨어
- ③ 프리웨어(Freeware) : 무료로 사용, 판매를 목적으로 재배포 금지(국내에서는 공개판)
- ④ 공개 소프트웨어(Public Domain) : 누구나 무료로 사용, 아무런 제약 없이 복제하여 영리 목적으로 배포도 가능

[저작권의 소유 형태에 따른 응용 소프트웨어의 배포]

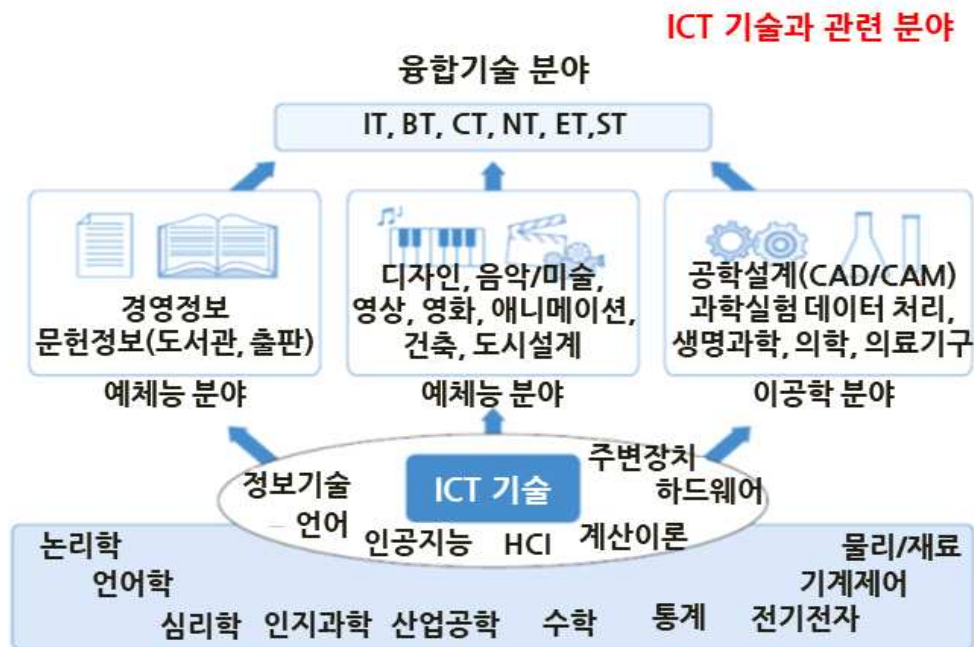


## 2. ICT 기술과 관련 학문 분야

### 1) ICT 기술의 기반 학문 분야

- ① 컴퓨터공학이나 컴퓨터 과학
  - 전자공학 및 전기공학
  - 물리학 및 재료공학
  - 화학이나 화학공학
  - 기계 및 제어 분야의 기술
  - 수학: 예) 알란 튜링, 폰 노이만, 애드가 카드 등
  - 언어학, 논리학, 심리학 등
  - HCI 분야: 인지과학, 심리학, 디자인학
  - 산업공학 및 통계학

② ICT 기술의 연계 학문 분야



【학습정리】

1. 소프트웨어 개발 방법론에는 폭포수 모델, 점진적 모델, 오픈소스 개발모델, 애자일 방법론, 컴포넌트 개발방법 등이 있다.
2. 객체지향 프로그래밍에서는 객체 또는 클래스를 하나의 컴포넌트로 간주한다.
3. 클래스다이어그램은 클래스간의 구조적 관계(재고기록, ←제품재고기록, ←부품재고기록)를 나타낸다.
4. 유리박스 테스트는 전체 소프트웨어에서 가장 중요하고 문제의 소지가 있는 일부 모듈을 중심으로 테스트하는 기법이다.