**03월 13일 수업 소프트웨어 공학 개론**

프로세스 모델  
개발과 운영을 통합해서 할 수 있는 시스템을 DevOps라고 한다.

[SDLC 절차 중간에 배포 추가됨]

[나선형 모델 그림 추가됨]

Unified Process => 각 단계마다 SDLC 절차를 모두 반복한다.  
점진적으로 수행한다. -> 처음에는 작게 시작해서 점점 넓여간다.  
시간이 지날수록 기획과 디자인 비중에서 개발, 구현 비중으로 마지막에는 배포와 테스트 비중으로 옮겨간다.

애자일(Agile)

기존 UP는 여러 단점이 있고 무겁기 때문에 애자일이 나오게 되었다.

프로세스, 도구보다는 개인과 상호작용이 더 중요하다.  
문서보다는 소프트웨어  
계약 협상보다는 고객과의 협력  
 => 갑과 을의 관계 향상

애자일 프로세스 기반의 다양한 방법론이 존재하는데, XP(eXtreme Programming) 등이 있다.  
페어 프로그래밍 -> 두 사람이 같은 코드를 보면서 의사소통하며 코드를 짜는 프로그래밍 방법.  
보통은 Scrum 기반을 사용한다. => UP는 아니기에 유연하게 대응한다.  
Kanban(간판) : 토요타가 생산을 효율적으로 관리하기 위해서 만든 시스템이다.   
 => 해야할 일들에 대한 목록을 만들어서 관리한다.  
 일에 대한 상태가 완료되면 다음 절차로 이동되는 시스템을 시각적으로 표현한다.

애자일 프로세스 특징  
 1. SW개발의 궁극적인 목표는 고객이 만족하는 소프트웨어를 개발하는 것이다.  
 2. 동작 가능한 소프트웨어를 짧으면 2주, 길어도 2개월 이내로 개발하고   
 고객의 피드백을 수용하여 개발한다.  
 3. 프로세스가 진행되는 동안 고객과 개발자가 지속적으로 소통한다.  
 => 그러나 약간의 위험요소는 존재한다.

객체지향 분석 및 설계 OOAD(Object Oriented Analysis and Design)

객체지향 : 사람이 현실세계 이해방법을 프로그램 개발에 적용한 방법이다.

Object : 물체, 대상 등 현실세계에 존재하는 모든 것들.

구체적 ~ 추상적 / 객체의 속성과 값 / 동작 : (뚜껑 열기), (마신다)

Object는 사람이 현실 세계 이해방법을 적용하기 위한 대상이며 속성과 동작이 있다.

Domain에 대한 이해

Domain : 소프트웨어를 이해하기 위한 대상 => SNS, Game, System 등 영역

속성이나 동작의 범위? + 상속  
상속 : 아기, 대학생  
사람 << 아기, 대학생-학번, 학과  
대학생 -> 학번, 학과  
추상적인 객체와 구체적인 객체와의 관계를 상속으로 구체화할 수가 있다.  
상속을 받음으로서 기초 특성을 이어받고 추가 특성을 정의할 수가 있다.   
 => 이러한 방식을 소프트웨어 개발에 추가하자.

동일한 동작이지만 (상속대상에 따라) 방법이 달라진다. => 다형성

Object와 Class의 차이점

Object는 상태(속성, 값)를 가지고 있다.  
Class는 Object를 만들 수 있는 틀  
Class로부터 만들어진 Object를 Instance라고 한다. (Instance = Object)

(플라톤의 국가에서 나온 단어들이다.)

[www.dinfree.com](http://www.dinfree.com)

한글로 표현 가능 & 관계도

대상을 선정하고 속성과 동작으로 만든다.  
어떤 속성이나 어떤 동작을 수행할지는 제한해서 만든다.

UML 스타일의 표기법  
Class 명, 속성, 동작

요즘은 Controller, Service, DTO 등 Detail은 별로 상관없다.

Domain Modeling

DDD 설계방식 => 요즘 많이 선호하는 설계 방식 중 하나이다.

동작이 없는 객체가 있을 수도 있다.

Data 구조 -> Entity 객체들은 동작이 없다.