

소프트웨어 공학

공학이란? : 수학과 자연과학 등의 학문을 이용하여 유용한 사물이나 환경을 구축하는 것을 목적으로 하는 학문

- 소프트웨어 공학? 소프트웨어의 개발, 운용, 유지보수 및 폐기에 대한 체계적인 접근 방법

- 소프트웨어 개발 생명주기 모델 : 소프트웨어 개발 생명주기모델 소프트웨어를 어떻게 개발할 것인가에 대한 전체적인 흐름. 주먹구구식, 폭포수, 프로토타이핑, 나선형 등이 있다.

- 폭포수모델 : 요구사항 분석 - 구조설계 - 구현 - 검증 - 유지보수

순차적 진행되기 때문에 시간이 끌리면 뒷부분이 수행되지 않는다.

앞 부분에서 생긴 오차가 뒷부분에서 큰 문제로 나타난다

- 프로토타이핑 모델 : 요구분석 - 프로토타입 설계 - 프로토타입 개발 - 프로토타입 평가 - 구현 - 인수 및 설치

프로토타입이 합격되었을 때 의욕이 떨어진다. 이미 어느 정도 완성 되었다고 생각하기 때문.

프로토타입 개발하는 시간과 구현하는 시간이 중복된다.

- 나선형모델 : 고객과의 소통 - 계획 및 정의 - 위험분석 - 구축 - 고객평가 - 고객과의 의사소통 - 계획 및 정의 - ...

고객이 원하는 것이 달라질 경우 소통이 지속되면서 처음 생각했던 것과 다른 결과물이 나올 수 있다.

- 소프트웨어 개발 방법론

소프트웨어를 생산하는데 필요한 반복적인 과정들을 정리한 것. 구조적 프로그래밍, 객체지향, 고속 개발 방법론, 익스트림 프로그래밍(Agile), 스트림(Agile), UP 등

- 애자일 개발 프로세스 : 특정 개발 방법론을 말하는 것이 아니라 기민하고 빠르게 개발하는 것을 가능하게 해주는 다양한 방법론 전체를 일컫는 말. 다른 고전적인 방법과 구별되는 가장 큰 차이점은 문서를 통한 개발방법이 아니라 실질적인 코딩을 통한 방법론이다.

어떤 회사가 '우리는 애자일하다' 라는 말을 썼을 때 '어떤 개발 프로세스를 가지고 있나요?' 라고 물어보면 더 좋은 대답을 들을 수 있다.

- UML (Unified Modeling Language) 통합 모델링 언어

표준화된 범용 모델링 언어. 객체지향 소프트웨어 집약 시스템을 개발할 때 산출물을 명세화, 시각화, 문서화 할 때 사용

클래스 모델, 클래스 다이어그램을 그리고 활용하는 것. 자바에서 유용하다

- TDD (Test driven development) 테스트 주도 개발

매우 짧은 개발 사이클을 반복하는 소프트웨어 개발 프로세스. 애자일과 쌍으로 나오는 단어이다.

결함을 점검하는 자동화된 테스트 케이스를 작성 - 케이스를 통과하기 위한 최소한의 양의 코드를 생성 - 새 코드를 표준에 맞도록 리팩토링

초반에는 테스트를 잘 하다가 후반에 일정에 쫓기게 되면 테스트 코드를 작성할 시간이 없어짐. 테스트코드를 작성하는 것 중요하나 실제로 수행하기는 무지 어려운 방법.

- PDD (Plan driven development) 계획 기반 개발

계획을 세우고 그 계획을 실천하는데 많은 시간과 노력을 할애하는 개발 방법. 고전적인 방법이다. 문서에 개발 관련 내용을 기재해 놓고 개발을 하는 방법.

- 형상관리

SW개발 및 유지보수 과정에서 발생하는 소스코드, 문서, 인터페이스 등 각종 결과물에 대해 형상을 만들고 이들 형상에 대한 변경을 체계적으로 관리, 제어하기 위한 활동

- 버전관리

형상관리의 일부. 일반적인 소프트웨어의 소스코드의 형상 만을 관리하는 것. SVN, Git, Mercurial 등

버전관리를 하는 이유는 책임소재에 있어 확실히 할 수 있기 때문이다. 그렇기 때문에 Git 을 활용할 때 커밋메시지를 잘 적어야 하는 것이다.

프로그래밍 언어

- 프로그래밍 언어의 종류

고급언어에서 저급언어로 변환되는 과정에 따른 분류

- 컴파일 언어 - C, C++, Go 등

실행파일이 만들어질 때 모든 번역을 해놓기 때문에 실행속도가 빠르나 배포 후 수정이 번거로움

- 인터프리터 언어 - BASIC, JavaScript, Python, Ruby

실행될 때 즉각 번역하는 방식으로 속도가 느리지만 수정이 쉬움

- 바이트코드 언어 - Java, C# 등

가상머신을 통해 번역을 하므로 사용자 환경에 영향을 덜 받음. 반대로 환경에 최적화 하기 어렵고 꼭 가상머신을 설치해야 한다는 단점이 있다.

프로그램을 번역하기 위해 가상머신이 필요하므로 리소스가 더 필요하다.

언어별 성능비교 참조 <https://attractivechaos.github.io/plb/>

0과 1이 해석되는 타이밍에 따라 분류하기도 하고 패러다임에 따른 분류를 하기도 한다.

객체지향 프로그래밍 패러다임

컴퓨터 프로그램을 명령어의 목록으로 보는 시각에서 벗어나 여러 개의 독립된 단위, 즉 객체들의 모임으로 파악하여 객체간의 상호작용으로 프로그램의 동작을 구현하고자 하는 것

- 클래스와 객체

클래스의 실질적인 형태 -> 객체

클래스 : 객체가 가질 수 있는 속성과 행위를 정의하는 틀

Car class가 있을 때 그 안에 색상, 기능, 인테리어설정 등이 들어갈 수가 있다. 거기서 Car object 가 나오면 서로 다른 색상, 같은 기능, 다른 인테리어 설정을 갖게 만들 수 있는 것이다.

프로그래밍 용어

- 개발자 Developer
- 서버 -> 데이터 제공 -> 클라이언트. 서버와 클라이언트는 상대적인 개념이다
- [Back end <-> Front end] <-> 사용자
- Thread : 프로세스 내에서 작업이 실행되는 흐름의 단위

Multi Thread : 한 프로세스 내에서 생성되는 여러 스레드는 공유된 메모리를 가지며 서로간의 자원을 공유하며 정보를 주고받는다. 프로그래밍을 통해 구현이 가능하다.

Multi Task : 정해진 Task를 처리하기 위해 하나의 프로세스가 운영체제의 스케줄링에 따라 조금씩 번갈아 가면서 수행하는 것

Multi Thread와 Multi Task의 차이에 대해서는 잘 알아둘 것.

- Library

특정기능을 수행할 수 있는 클래스 또는 함수의 집합체

수학 라이브러리, 애니메이션 라이브러리, 문자열 라이브러리 등

- API Application Programming Interface

응용 소프트웨어 프로그래밍 접합부

응용 소프트웨어 프레임워크 사이의 중간매체

소프트웨어 간의 통신을 위해 메시지를 전달하는 방식 등이 결정된 것

‘너네 앱에서 필요한 정보를 내가 주는 창구를 마련해줄게!’

- Framework

구조적으로 고정된 부분을 재사용할 수 있도록 하고, 응용별 특정 기능을 추가적인 사용자 작성 코드에 의해 선택적으로 구현가능 하도록 하는 포괄적인 추상 구조, 그리고 이를 지원하는 소프트웨어 플랫폼

기반으로 깔아두는 기능들을 제공하고 그 위에 추가 구현 할 수 있게 한다.

- 비유하자면

프레임워크 – 국가(정부)

프레임워크의 하위 프레임워크 – 정부부처

API - 국가기관

라이브러리 - 직무전문가

프로그래밍 언어 - 국가 통용 언어

레퍼런스 문서 - 국가 조직도 및 기능 설명서

애플에서는 IOS에서 이용할 수 있게 Cocoa Touch Framework 를 제공한다.

Cocoa touch 라는 나라에 내가 원하는 일을 할 수 있는 기업을 세우는 것이다.

Cocoa touch 라는 나라에서 Objective-C, Swift 라는 언어를 통해 의사소통한다.

- 디자인 패턴

프로그램 개발에서 자주 나타나는 과제를 해결하기 위한 방법 중 하나.

과거의 소프트웨어 개발 과정에서 발견된 설계의 노하우에 이름을 붙여 이후에 재이용하기 좋은 형태로 묶어서 정리한 것

MVC (Model View Controller) : 데이터 관리와 화면표시를 따로 관리하며 그 둘 사이를 처리해주는 중재자가 있는 형태의 클래스 디자인

그 외 MVVM (Model View ViewMotel), Observer, Singleton, Prototype 등

- Reference Document

API에 대해 서술해 놓은 문서

- IDE (Integrated Development Environment)

- SDK (Software Development Kit) 소프트웨어 개발에 필요한 도구의 모음

SDK = IDE + Framework + Tools + ...