

시스템분석설계 개인 포트폴리오

20214212 이효준

목차

1. 소프트웨어 공학

2. 깃과 깃허브

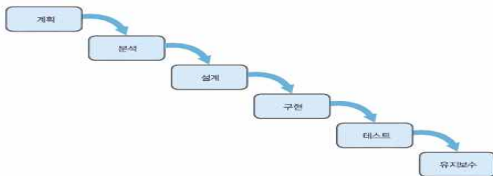


1. 소프트웨어 공학



1-1. 소프트웨어 공학 개요

- 소프트웨어 공학 = 소프트웨어의 개발, 운용, 유지보수 등의 생명 주기 전반을 체계적이고 서술적이며 정량적으로 다루는 학문이다. 즉, 공학을 소프트웨어에 적용하는 것이다.
- 소프트웨어 특징 4가지
 - 복잡성 - 순응성 - 변경성 - 비가시성
- 소프트웨어 개발 생명주기 = 계획에서 유지보수 단계에 이르기까지 일어나는 일련의 과정



1-2. 소프트웨어 개발 프로세스

- 1단계: 계획
- 2단계: 요구분석
- 3단계: 설계
- 4단계: 구현
- 5단계: 테스트
- 6단계: 유지보수

1단계: 계획

- 개발 비용 산정, 일정 계획(작업분할구조도 사용)
위험 관리

2단계: 요구 분석

- 기존 시스템의 문제점 파악-> 새로운 요구사항도출
-> 다이어그램 작성
- 개발 방법론에 따른 표현 도구
(구조적 방법론: DFD, DD, Mini Spec
정보공학 방법론: E-R 다이어그램
객체지향 방법론: UML의 유스케이스 다이어그램
- 최종 산출물: 요구 분석 명세서

1-2. 소프트웨어 개발 프로세스

3단계: 설계

- 설계 원리: 분할과 정복, 추상화, 단계적분해, 모듈화, 정보은닉
- 소프트웨어 아키텍처, 객체지향 설계, 아키텍처 스타일
- GoF의 디자인 패턴
- 모듈 평가기준: 응집도와 결합도

4단계: 구현

5단계: 테스트

- 개발자 또는 사용자 시각에 따른 분류
- 사용되는 목적에 따른 분류
- 품질 특성에 따른 분류
- 소프트웨어 개발단계에 따른 분류

6단계: 유지보수

- 수정 유지보수, 적응 유지보수, 기능보강 유지보수, 예방 유지보수

2. 깃(Git)과 깃허브(GitHub)



git



GitHub

2-1. 깃(Git)

- 깃(Git)은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 분산 버전 관리 시스템이다
- 소프트웨어 개발에서 소스 코드 관리에 주로 사용되지만 어떠한 집합의 파일의 변경사항을 지속적으로 추적하기 위해 사용될 수 있다
- 분산 버전 관리 시스템으로서 빠른 수행 속도에 중점을 두고 있는 것이 특징이며 데이터 무결성, 분산, 비선형 워크플로를 지원한다

깃(Git)의 장점

- 빠르다
- 모든 작업자가 원본을 가지고 있다
- 분산 버전 관리 시스템
- Local에서 대부분의 작업을 할 수 있다

2-2. 깃허브(GitHub)

- 깃허브(GitHub)는 분산 버전 관리 툴인 깃 저장소 호스팅을 지원하는 웹 서비스이다
- GitHub는 영리적인 서비스와 오픈소스를 위한 무상 서비스를 모두 제공한다.
- 깃이 텍스트 명령어 입력 방식인데 반해, 깃허브는 그래픽 유저 인터페이스(GUI)를 제공한다

깃허브(GitHub)의 서비스들

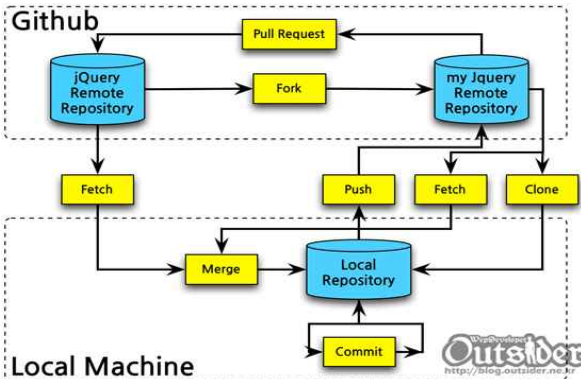
- 저장소
- 깃허브 페이지
- 지속적 통합 서비스인 깃허브 액션
- 패키지 저장소인 깃허브 패키지



2-2. 깃허브(GitHub)

o Fork

- 다른 사람의 github repository를 복제하여 내가 편리하게 쓸 수 있도록 나의 github repository로 복제하는 기능
- Fork한 저장소는 연결되어 있다



THANK YOU

지금까지 발표들어주셔서 감사합니다