시스템 분석 설계
개인 포트폴리오

컴퓨터정보공학과 PB 202142O7 최 지혜

목차

1 소프트웨어공학의 이해

2 깃과 깃허브

3 Web Site 제작 절차

Part 1

소프트웨어공학의 이해

Part 1

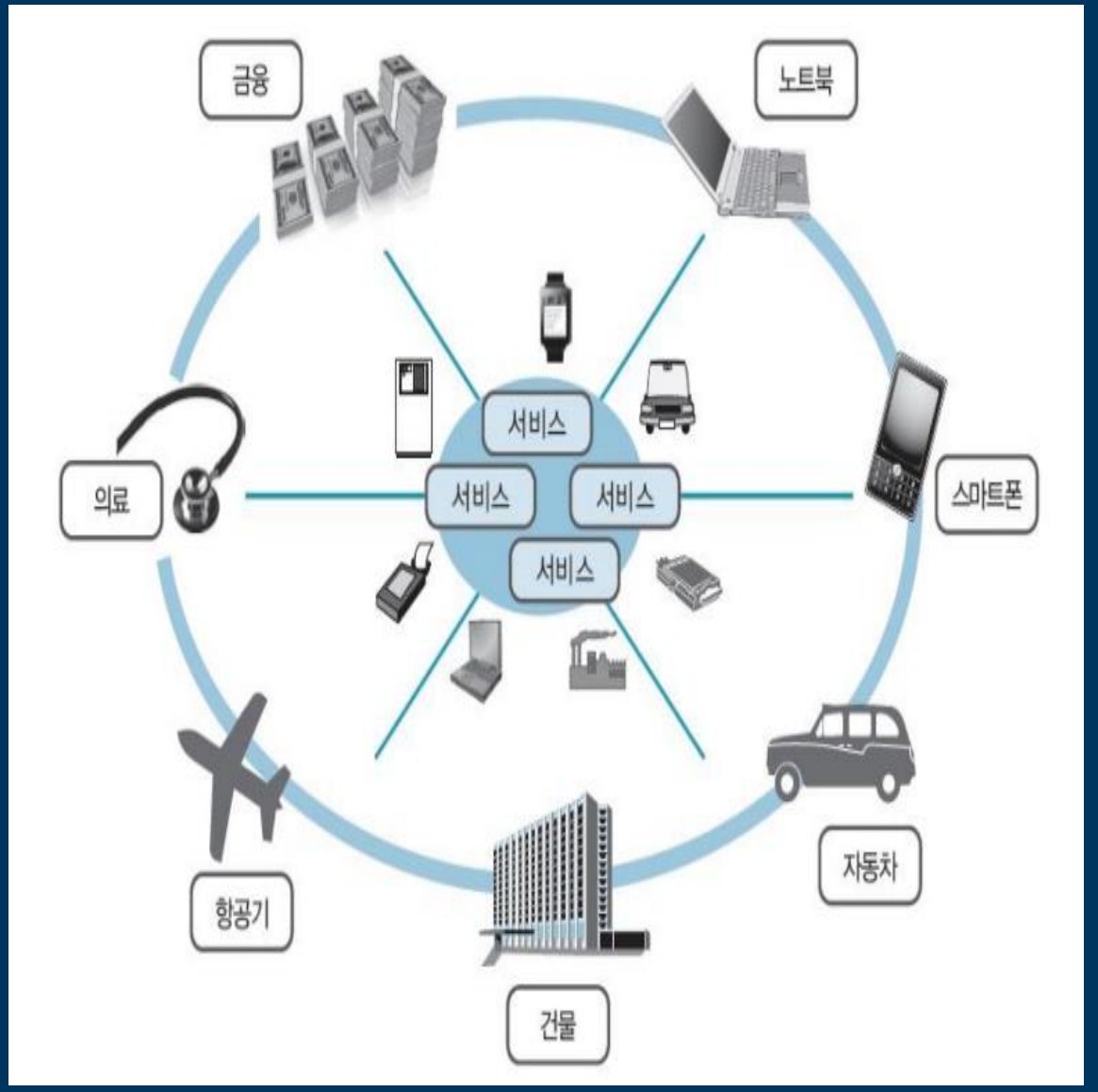
소프트웨어 공학이란?

정의

- 소프트웨어 + 공학
- 소프트웨어의 설계, 제작, 신뢰성 등을 공학적 원리를 적용하여 소프트웨어를 연구하는 학문.

목적

- S/W 개발의 어려움을 해결
- 효율적인 개발을 통한 생산성을 향상
- 고품질 소프트웨어 제품을 생성



Part 1

소프트웨어의 특징

제조가 아닌 개발

- 제조는 정해진 틀에 맞춰 일정하게 생산하여 능력에 따른 결과물의 차이가 크지 않다.
- 개발은 개인 능력에 따라 결과물과 생산성의 차이가 크다.

소모가 아닌 품질 저하

- H/W는 오래 사용하면 부품도 닳고, 고장 발생 빈도도 높아지며, 기능도 떨어진다.
- S/W는 오래 많이 사용한다 해도 닳지 않고, 고장 빈도도 낮으며, 시간이 지날수록 좋은 소프트웨어가 개발되어 사용 빈도가 줄어드는 것이지 기능은 동일하다.

Part 1

소프트웨어 개발 생명주기 (SDLC)

1단계: 계획

- 개발 비용 산정 : COCOMO모델, 기능점수(FP)모델 사용
- 일정 계획 : 작업분할구조도(WBS), CPM 사용

2단계 : 요구분석

- 기존 시스템의 문제점을 파악하여 요구사항을 도출해서 다이어그램을 작성한다
- 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

Part 1

소프트웨어 개발 생명주기 (SDLC)

3단계: 설계

- 설계 원리 : 분할과 정복, 추상화, 단계적 분해, 모듈화, 정보은닉
- 소프트웨어 아키텍처, 객체지향 설계
- 아키텍처 스타일
- GoF의 디자인 패턴
- 모듈 평가 기준 : 응집도와 결합도

4단계 : 구현

- 간략한 프로그래밍 언어의 역사
- 표준 코딩 규칙

Part 1

소프트웨어 개발 생명주기 (SDLC)

5단계 : 시험

- 테스트의 절차
- 개발자 또는 사용자 시각에 따른 분류
- 사용되는 목적에 따른 분류
- 품질 특성에 따른 분류
- 소프트웨어 개발 단계에 따른 분류

6단계 : 유지보수

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

Part 2

소프트웨어 프로세스 모델

- 주먹구구식 모델
- 선형 순차적 모델(폭포수 모델)
- V 모델
- 진화적 프로세스 모델(프로토타입 모델)
- 나선형 모델
- 단계적 개발 모델
- 통합 프로세스 모델(UP)
- 애자일 프로세스 모델

Part 2

깃과 깃허브

Git이란?

- 버전 관리 시스템(VCS ; Version Control System)으로 파일 변화를 시간에 따라 기록했다가 나중에 특정 시점의 버전을 다시 꺼내 올 수 있는 시스템

Github란?

- 클라우드 방식으로 관리되는 버전 관리 시스템(VCS)
- 자체 구축이 아닌 빌려쓰는 클라우드 개념을 말한다.
- 오픈소스는 일정 부분 무료로 저장이 가능하고 아닌 경우에는 유료로 사용해야 한다.

ⓒSaebyeol Yu. Saebyeol's PowerPoint

Part 3

버전 관리

버전 관리란?

- 소스 하나 또는 묶음을 하나의 버전으로 간주한다.
- 파일이나 폴더를 추가, 수정, 삭제하며 사람이 직접 관리한다.
- 원할 때 예전 버전 내용 전체를 되돌려 볼 수 있으며 복잡한 코드를 개발할 때 이전 버전과 비교하기가 쉽다.

버전 관리가 필요한 이유

- 개발자 사이의 협업에 필요하다.
- 전체 개발 소스를 공유하면서 개발 파트를 나눌 수도 있고, 같은 모듈을 개발하더라도 소스를 서로 공유하며 개발 할 수 있다.

Part 3

Git 관련 주요 용어

- 저장소 (Repository)
- 소스코드가 저장되어 있는 여러 개의 브랜치들이 모여 있는 디스크상의 물리적인 공간을 의미한다.

- 체크 아웃 (Check out)
- 특정 시점이나 브랜치의 소스코드로 이동하는 것을 의미한다.
- 체크아웃 대상은 브랜치, 커밋 그리고 태그이다.
- 체크아웃을 통해 과거 여러 시점의 소스코드로 이동할 수 있다.

- 스테이지 (Stage)
- 작업한 내용이 올라가는 임시 저장 영역이다.
- 이 영역을 이용하여 작업한 내용 중 커밋에 반영할 파일만 선별하여 커밋을 수행할 수 있다.

- 커밋 (Commit)
- 작업한 내용을 로컬 저장소에 저장하는 과정이다.

- 태그(Tag)
- 커밋의 임의 위치에 쉽게 찾아갈 수 있도록 붙여 놓은 이정표를 태그라고 한다.

Part 3

Git 관련 주요 용어

- 푸시 (Push)
- 로컬 저장소의 내용 중 원격 저장소에 반영되지 않을 커밋을 원격 저장소로 보내는 과정이다.

- 풀 (Pull)
- 푸시와 반대로 원격 저장소에 있는 내용 중 로컬 저장소에 반영되지 않은 내용을 가져와서 로컬 저장하는 과정을 의미한다. 이를 통해 다른 팀원이 변경하고 푸시한 내용을 로컬 저장소로 가져올 수 있다.

- 브랜치 (Branch)
- 커밋을 단위로 구분된 소스코드 타임라인에서 분기해서 새로운 커밋을 쌓을 수 있는 가지를 만드는 것, 혹은 그 가지를 브랜치라고 한다.

- 병합 (Merge)
- 브랜치와 반대되는 개념으로, 하나의 브랜치를 다른 브랜치와 합치는 과정을 의미한다..

Part 3

Web site 제작 절차

Web Site 제작이란,

웹 사이트 개발 과정을 포함하여 웹 사이트의 기획과 출시 단계까지 아우르는 개념이다. 웹 사이트 제작은 일반적인 응용 소프트웨어 제작과 같은 단계(서비스 기획 – 디자인 – 개발 – 테스트 – 출시 및 배포)를 거친다.

Part 3

1. 서비스 기획

- 아이디어 구체화
- 어떤 사용자를 타겟으로 할 것인지, 사용자의 니즈는 무엇인지, 제공할 가치는 어떤 것인지 그리고 최종 목표는 무엇인지를 정리한다.

- 시나리오 정리
- 사용자가 앱을 사용하는 시나리오를 정리한다. 타겟별로 시나리오를 작성하거나 시간대별, 지역대별로 구분해서 작성할 수도 있다.

- 구체적인 기능의 리스트 작성
- 시나리오 별로 필요한 기능 리스트를 작성한다.

- 프로젝트 인원 구성
- 앱 개발은 보통 여러명이 협업하여 진행한다. 프로젝트에 필요한 역할은 기획자, 개발자, 디자이너 이다. 추가로 프로젝트 매니저, 테스터를 따로 두기도 한다.

Thank you