

시스템 분석 설계

개인 포트폴리오

20181787 이혁주

2022-05-31



목차



1

소프트웨어 공학

2

Git & GitHub

3

Node JS

4

MySQL

소프트웨어 공학



01°

소프트웨어 공학 – 소프트웨어의 개요

- 소프트웨어는 하드웨어를 동작 시켜 사용자가 작업을 편하게 수행하도록 하는 프로그램과 자료구조 등을 총칭.
- 소프트웨어는 프로그램 자체 뿐만 아니라 프로그램의 개발, 운용 및 유지보수에 관련된 모든 문서와 정보를 포함.

소프트웨어 공학 – 소프트웨어의 특징

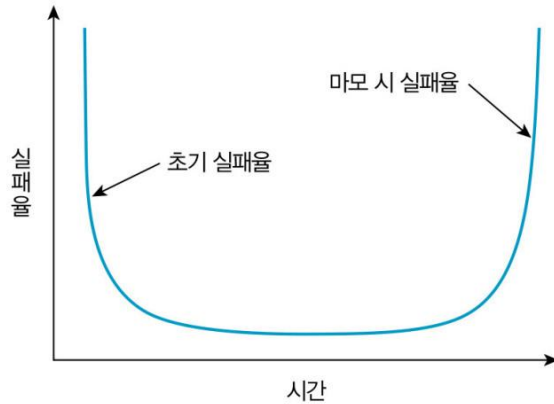
- **제조가 아닌 개발**

제조는 정해진 틀에 맞춰 일정하게 생산하는 것을 말하지만 소프트웨어는 개인간의 능력차이에 따라 나오는 결과물이 다름.

- **소모가 아닌 품질 저하**

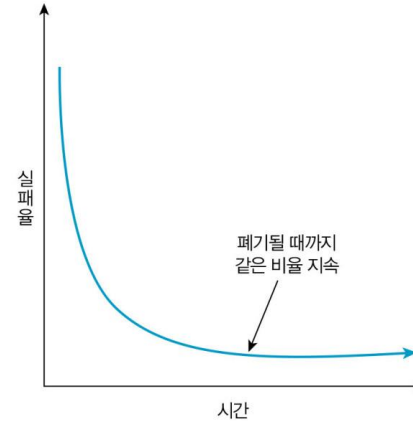
하드웨어의 경우 부품이 닳고 기능이 떨어지지만 소프트웨어는 물건처럼 닳는게 아니라 시간이 지나 품질의 저하가 오는 것.

소프트웨어 공학 – 소프트웨어의 실패 곡선



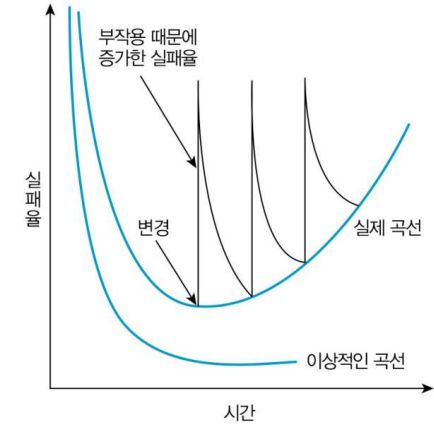
H/W의 실패곡선

- 초기 실패율 높음
- 오류 해결
- 오랜 기간 사용하며 문제 발생
- 실패율 다시 증가
- U자형 곡선을 띰



S/W의 이상적 실패곡선

- 발견되지 않은 오류로 실패율 높음
- 오류 해결
- 오랜 기간 사용
- L자형 곡선을 그림



S/W의 실제실패곡선

- 이상적인 곡선과 다르게 업데이트마다 실패율이 급격히 증가
- 오류를 해결
- 업데이트마다 순차적으로 반복됨

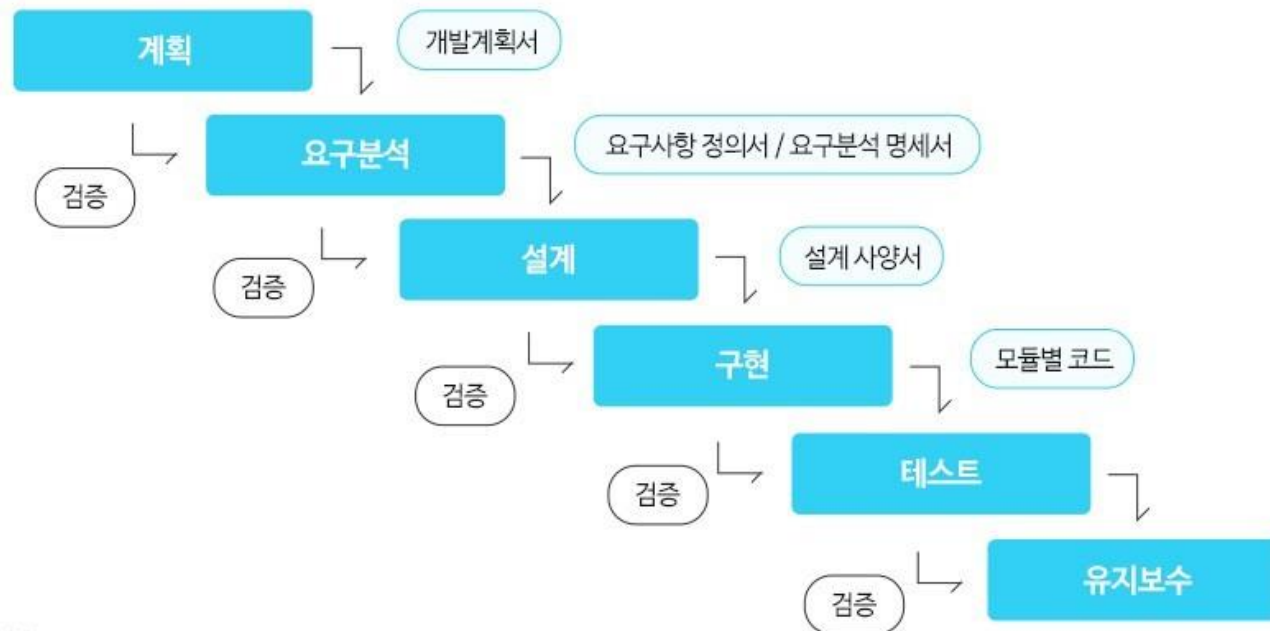
소프트웨어 공학 – 소프트웨어의 개발속도

- 하드웨어보다 소프트웨어의 개발 속도가 느림
근본적인 개발 방법의 차이
- 하드웨어 : 검증 받은 부품을 조립하는 형태
- 소프트웨어 : 처음부터 만들어가는 개발 형태

소프트웨어 공학 – 소프트웨어의 개발 생명 주기(SDLC)

- 폭포수모델(Waterfall Model)

정해진 단계 선형 순차적 모델, 분석, 설계, 구현, 시험 및 유지보수 과정



소프트웨어 공학 – 작업분할구조도(WBS)

- **작업 분할 구조도**

프로젝트 목표 달성을 위하여 필요한 업무들과 액티비티들을 세분화 하는 작업

계층 구조 중 최하위에 있는 항목은 작업 패키지라고 하며, 이 작업 패키지는 담당자를 할당할 수 있을 정도로 작게 나뉘어야 함

Git & GitHub



02°

Git & GitHub – Git & GitHub가 무엇인가

Git

- 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율하기 위한 형상 관리 도구
- git을 통해 파일의 변경사항을 확인 및 적용하고 그 차이점을 확인할 수 있음.

GitHub

- Git을 위한 웹 저장소
- 개발자들의 커뮤니티 협의(협업) 공간
버전 관리를 위한 서버 저장소 및 프로젝트 개발을 위한 협업 관리 서비스

Git & GitHub – GitHub 용어

- **Fork** : 내 계정의 저장소에 다른 사람의 저장소를 복사하는 것
- **Upstream** : GitHub의 모든 오픈 프로젝트 저장소
- **Origin** : 나의 저장소(GitHub의 내 계정에 있는 Fork한 저장소도 포함)
- **로컬 저장소** : PC에 복사한 지역 저장소
- **Push**: 로컬 저장소의 프로젝트를 오리진에 업데이트할 때
- **Fetch(Pull이라고도 함)** : Origin에서 로컬 저장소로 가지고 올 때
- **Fetch & Merge** : Upstream에서 Origin으로 끌어올 때
- **Clone** : 주인이 내 것인 저장소를 로컬 저장소에 끌어올 때
(Fork와 차이점은 소유자가 다른 사람의 것)

Node JS



03°

NodeJS – Node JS란?

- **Node JS** : 자바스크립트 엔진 **V8** 위에서 동작하는 이벤트 처리 **I/O** 프레임워크 서버 환경에서 자바스크립트로 애플리케이션을 작성할 수 있도록 도와줌.
- **장점 1 : 비동기 프로그래밍**
동기 프로그래밍은 무엇인가 요청하면 결과를 즉시 받는 것을 의미함.
비동기는 이벤트를 요청하고 바로 결과를 받지 않아도 됨.
따라서 결과값을 기다리지 않고 보다 다양한 요청을 처리할 수 있음.
웹 분야에서는 비동기 프로그래밍을 쓰는 경우가 드물었는데, **Node JS**로 비동기 프로그래밍을 비교적 쉽게 할 수 있게 됨.
- **장점 2 : Front-End와 Back-End를 하나의 언어로 다같이 관리할 수 있음**
덕분에 웹 개발자들은 새로운 언어를 배우지 않고 기존 언어를 활용해 서버 기술을 빨리 응용하게 됐다. 확장성도 **Node JS**의 장점이다.
- **단점** : 관련 예제나 모듈의 완성도 등이 타 언어에 비해서는 조금 부족함

NodeJS – 예제 코드

- 메인 인덱스 코드

```
//Module Require
const express = require("express"),
      app = express(),
      bodyParser = require("body-parser"),
      api = require("./api/index")

//Middleware Set
//body-parser을 이용한 요청 데이터 처리
app.use(bodyParser.json())
app.use(bodyParser.urlencoded({ extended: false }))

// /api 경로에 요청이 올 경우 api로 권한을 넘김"
app.use("/api", api)

//Port Set
const port = 8001

//Server On
app.listen(port, () => console.log(`SERVER ON! PORT : ${port}`))
```

NodeJS – 예제코드

- DB 연결 코드

```
//Module Require
const mysql = require("mysql"),
    //DB 정보 입력
    db_info = {
        host: "dms-pine-ec2-4nzmj-ep-northeast-2-rds.amazonaws.com", //DB URL
        port: "3306", //DB PORT
        user: "root", //DB 관리자 계정
        password: "qwer1595", //DB 관리자 계정 PW
        database: "HAT", //DB 선택
    },
    pool = mysql.createPool(db_info)

module.exports = (callback) => {
    pool.getConnection((err, conn) => {
        if (!err) {
            callback(conn)
        }
    })
}
```


MySQL



04°

MySQL – 데이터 베이스란

- 데이터 베이스
여러 사람에 의해 공유되어 사용될 목적으로 통합하여 관리되는 데이터의 집합
- 데이터 베이스의 특성
 - 똑같은 자료를 중복하여 저장하지 않는 통합된 자료
 - 컴퓨터가 액세스하여 처리할 수 있는 저장장치에 수록된 자료
 - 어떤 조직의 기능을 수행하는 데 없어서는 안 되며 존재 목적이 뚜렷하고 유용성 있는 운영 자료
 - 한 조직에서 가지는 데이터베이스는 그 조직 내의 모든 사람들이 소유하고 유지하며 이용하는 공동 자료로서 각 사용자는 같은 데이터라 할지라도 각자의 응용 목적에 따라 다르게 사용할 수 있다는 점

MySQL – MySQL이란

- **1995년에 발표된 오픈 소스 RDBMS**
- **MySQL Workbench라는 GUI 도구를 이용하여 개발할 수 있음.**
- **오라클 DB, MS SQL, PostgreSQL과 함께 세계 4대 RDBMS 자리를 차지하고 있음**

MySQL – MySQL 기초 문법 - DATABASE

- 데이터 베이스 생성 : **CREATE DATABASE** (데이터베이스 명);
- 데이터 베이스 보기 : **SHOW DATABASES;**
- 데이터 베이스 사용하기 : **USE** (데이터베이스 명);
- 데이터 베이스 삭제하기 : **DROP** (데이터베이스 명);

MySQL – MySQL 기초 문법 – TABLE

- 테이블 생성 : **CREATE TABLE** (데이터베이스 명)
(컬럼명1 데이터타입, 컬럼명2 데이터타입...);
- 테이블 보기 : **SHOW TABLES;**
- 특정 테이블 구조 보기: **DESC** (테이블 명);

MySQL – MySQL 기초 문법 – CRUD(1)

- **CREATE**

**INSERT INTO (테이블명)(컬럼명1, 컬럼명2, ...) VALUES
('VALUE1' , ' VALUE2' ,...)**

- **READ**

**SELECT (컬럼 명) FROM (테이블명) WHERE (조건) LIMIT (조회할 레코드 수)
ORDER BY (DESC or ASC)**

MySQL – MySQL 기초 문법 – CRUD(2)

- **UPDATE**

UPDATE (테이블명) SET (컬럼명) = '변경할 값' WHERE (조건)

- **DELETE**

DELETE FROM (테이블명) WHERE (조건)

*** WHERE 절이 없으면 테이블의 모든 데이터가 삭제됨**

감사합니다

