

# [2022 시스템 분석 설계]

## 포트폴리오

컴퓨터정보공학과 2-B 20202831 김유림



# Index

## 01

### 소프트웨어 공학

- 소프트웨어 이해
- 소프트웨어의 특징
- 이상적인 소프트웨어 실패곡선과  
실제 소프트웨어 실패곡선
- 소프트웨어 개발단계
- 소프트웨어 공학

## 02

### 깃과 깃허브

- 깃과 깃허브
- 깃허브 용어

## 03

### Arduino

- 작업분할구조도
- 주요기술

# 01 소프트웨어 공학

## 01. 소프트웨어 공학

### 1) 소프트웨어 이해

# 01 소프트웨어 이해

## 1. 프로그램과 소프트웨어의 정의

### · 프로그램

컴퓨터에서 실행될 때 특정 작업을 수행하는 일련의 명령어들의 모음 (집합체)이다.

원시 코드인 C언어, JAVA, Python 등의 언어로 쓰인다.

### · 소프트웨어

응용프로그램과 데이터처럼, 컴퓨터의 하드웨어상에서 구동되거나 처리하는 무형물을 지칭하는 말이다.

원시코드, 모든 산출물(자료구조, DB구조 등) 그리고 각 단계마다 생산되는 문서 등이 있다.

▶ 소프트웨어는 프로그램 뿐만 아니라 그 이상의 것도 포함하는 매우 포괄적인 개념이다.

## 01. 소프트웨어 공학

### 2) 소프트웨어의 특징

# 01 소프트웨어 이해

## 2. 소프트웨어의 특징

소프트웨어의 4가지 대표 특징(복잡성, 순응성, 변경성, 비가시성)

### ·복잡성

자동화하려는 대상이 복잡할 뿐아니라 수많은 내부 요소들로 이루어져 상호작용이 많고 복잡하다.

### ·순응성

사용자의 요구나 환경, 데이터 변화에 따라 적절히 변형되는 특징이 있다.

## 01. 소프트웨어 공학

### 2) 소프트웨어의 특징

# 01 소프트웨어 이해

## 2. 소프트웨어의 특징

### ·변경성

문자로 구성된 프로그램이므로 쉽게 변경할 수 있고 개발 과정에 따라 자주 변경된다. 하지만 복잡하고 규모가 커지면 간단한 수정이라도 관련된 요소를 연쇄적으로 변경해야 하므로 어려운 작업이 될 수도 있다.

### ·비가시성

종이나 자기 디스크와 같은 유형 매체에 저장하지만, 개념적이고 무형적이다. 빌딩이나 자동차는 그 생산물을 보고 그 구조를 쉽게 파악할 수 있지만 소프트웨어는 그 구조가 코드 안에 숨어 있어서 쉽게 드러나지 않는다.

## 01. 소프트웨어 공학

### 3) 이상적인 소프트웨어 실패곡선과 실제 소프트웨어 실패곡선

# 01 소프트웨어 이해

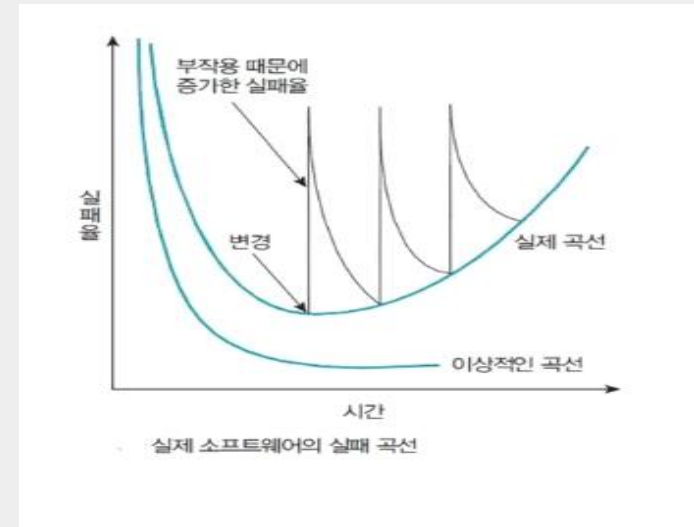
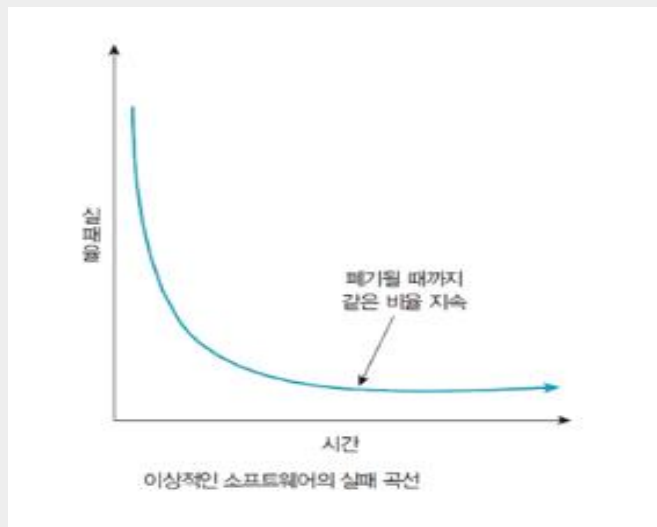
## 3. 이상적인 소프트웨어 실패곡선과 실제 소프트웨어 실패 곡선

### · 이상적인 소프트웨어 실패곡선

발견되지 않은 오류들로 인해 초기 실패율이 높지만 오류해결후에는 오랜기간동안 문제없이 사용가능하다.

### · 실제 소프트웨어 실패곡선

개발완료 후 변경 사항(기능 추가 및 수정)과 환경 변화로 인한 부작용으로 실패율이 급격히 증가한다.



## 01. 소프트웨어 공학

### 4) 소프트웨어 개발단계

# 01 소프트웨어 이해

## 4. 소프트웨어 개발단계 (6단계로 구성)

계획 > 요구분석 > 설계 > 구현 > 테스트 > 유지보수

### 1단계 계획

#### 1) 개발 비용 산정(COCOMO모델, 기능점수모델 사용)

##### · COCOMO모델

비용추정 모델로서 소프트웨어 개발비, 유지보수의 비용 견적, 개발 단계 및 업무 활동 등 비용 견적의 유연성이 높아 가장 널리 사용된다.

##### · 기능점수모델

소프트웨어의 생산성을 측정하기 위해 개발되었다.



## 01. 소프트웨어 공학

### 4) 소프트웨어 개발단계

# 01 소프트웨어 이해

## 4. 소프트웨어 개발단계 (6단계로 구성)

계획 > 요구분석 > 설계 > 구현 > 테스트 > 유지보수

### 1단계 계획

#### 2) 일정계획

작업분할구조도와 CPM을 사용한다.

- 작업분할구조도(WBS)

프로젝트 관리 및 시스템 엔지니어링에서 프로젝트를 소규모의 구성단위까지 나누어 전달할 수 있는 기능을 말함

- CPM

캠페인 기간 동안 일어난 노출 1,000회당 지불하는 비용을 뜻한다.

## 01. 소프트웨어 공학

### 4) 소프트웨어 개발단계

# 01 소프트웨어 이해

## 4. 소프트웨어 개발단계 (6단계로 구성)

계획 > 요구분석 > 설계 > 구현 > 테스트 > 유지보수

### 2단계 요구분석

개발 방법론에는 구조적 방법론, 정보공학 방법론, 객체지향 방법론으로 총 3가지의 표현도구가 있다.

기존 시스템의 문제점을 파악 > 새로운 요구사항 도출 > 다이어그램 작성

위 방법을 따라 최종적으로는 요구 분석 명세서가 생성된다.

## 01. 소프트웨어 공학

### 4) 소프트웨어 개발단계

# 01 소프트웨어 이해

## 4. 소프트웨어 개발단계 (6단계로 구성)

계획 > 요구분석 > 설계 > 구현 > 테스트 > 유지보수

### 3단계 설계

설계의 원리는 분할과 정복, 추상화, 단계적 분해, 모듈화, 정보은닉에 따른다.

SW설계는 표준화가 잘 되어있지 않다. 따라서 설계도면만으로는 제대로 구현하기 힘들다.

소프트웨어 아키텍처, 객체지향 설계

아키텍처 스타일

GoF의 디자인 패턴

모듈의 평가는 응집도와 결합도에 따라 평가된다.

## 01. 소프트웨어 공학

### 4) 소프트웨어 개발단계

# 01 소프트웨어 이해

## 4. 소프트웨어 개발단계 (6단계로 구성)

계획 > 요구분석 > 설계 > 구현 > 테스트 > 유지보수

### 4단계 구현

간략한 프로그래밍 언어의 역사, 표준 코딩 규칙

### 5단계 테스트

테스트의 절차를 통해 문제점을 파악한다.

개발자 또는 사용자 시각에 따른 분류, 사용되는 목적에 따른 분류, 품질 특성에 따른 분류, 소프트웨어 개발에 따른 분류등을 통해 테스트에서의 문제점을 조금 더 정확하게 파악할 수 있다.

### 6단계 유지보수

수정, 적응, 기능보강 및 예방으로 총 4가지의 유지보수를 한다.



## 01. 소프트웨어 공학

### 5) 소프트웨어 공학

# 01 소프트웨어 이해

## 5. 소프트웨어 공학

### · 정의

소프트웨어 + 공학의 의미로 소프트웨어의 개발, 운영, 유지보수 및 폐기에 대한 체계적인 접근 방법을 연구하는 학문이다.

### · 특징

소프트웨어 개발 전 과정에 걸쳐 필요한 이론, 개념 및 기술을 다룬다.  
소프트웨어 개발 과정에서 생산되는 모든 산출물이 그 대상이 된다.

### · 목적

SW 개발의 어려움을 해결한다.  
효율적인 개발을 통해 생산성을 향상시킨다.  
고품질 소프트웨어 제품을 만들 수 있다.

## 02 깃과 깃허브

## 02. 깃과 깃허브

### 1) 깃과 깃허브

# 02 깃과 깃허브

## 1. 깃과 깃허브

### · 깃(Git)

깃은 컴퓨터 파일의 변경사항을 추적하고 여러 명의 사용자들 간에 해당 파일들의 작업을 조율할 수 있는 형상 관리 도구이다. 리눅스의 창시자인 리누스 토발즈가 리눅스 커널 프로젝트를 위한 분산 버전 관리 시스템으로 개발하였다.

### · 깃허브(GitHub)

깃허브는 분산 버전 컨트롤 소프트웨어 깃을 기반으로 소스 코드를 호스팅하고, 협업 지원 기능들을 지원하는 마이크로소프트의 웹서비스이다.

자체 구축이 아닌 빌려쓰는 클라우드의 개념이다.

오픈 소스는 일정 부분 무료로 저장 가능하지만 유료로 사용하는 부분도 있다.

## 02. 깃과 깃허브

### 2) 깃허브 용어

## 02 깃과 깃허브

### 2. 깃허브 용어

- **fork**

원격 저장소를 로컬 저장소로 복제한 뒤 새로운 원격 저장소에 푸시하는 과정이 fork이다.

- **upstream**

타인의 (오픈소스 프로젝트)의 깃허브 저장소를 뜻할 때 사용하는 용어이다.

- **origin**

나의 깃허브(github에서 내 계정에 있는 fork한 저장소)를 뜻할 때 사용하는 용어이다.

- **로컬 저장소**

내가 나의 컴퓨터에 복사한 지역 저장소를 뜻한다.



## 02. 깃과 깃허브

### 2) 깃허브 용어

## 02 깃과 깃허브

### 2. 깃허브 용어

#### · Commit

작업한 내용을 로컬 저장소에 저장하는 과정이다.  
각각의 커밋은 의미있는 변경 단위이고, 변경에 대한 설명을 커밋 로그로 남긴다.

#### · Push

로컬 저장소의 내용 중 원격 저장소에 반영되지 않을 커밋을 원격 저장소로 내보내는 과정이다.

#### · Pull

푸시와 반대되는 용어로 원격 저장소에 있는 내용 중 로컬 저장소에 반영되지 않은 내용을 가져와 로컬 저장소에 저장하는 과정을 의미한다.

#### · 스테이지

작업한 내용이 올라가는 임시 저장 영역이다.

TM

# 03 Arduino

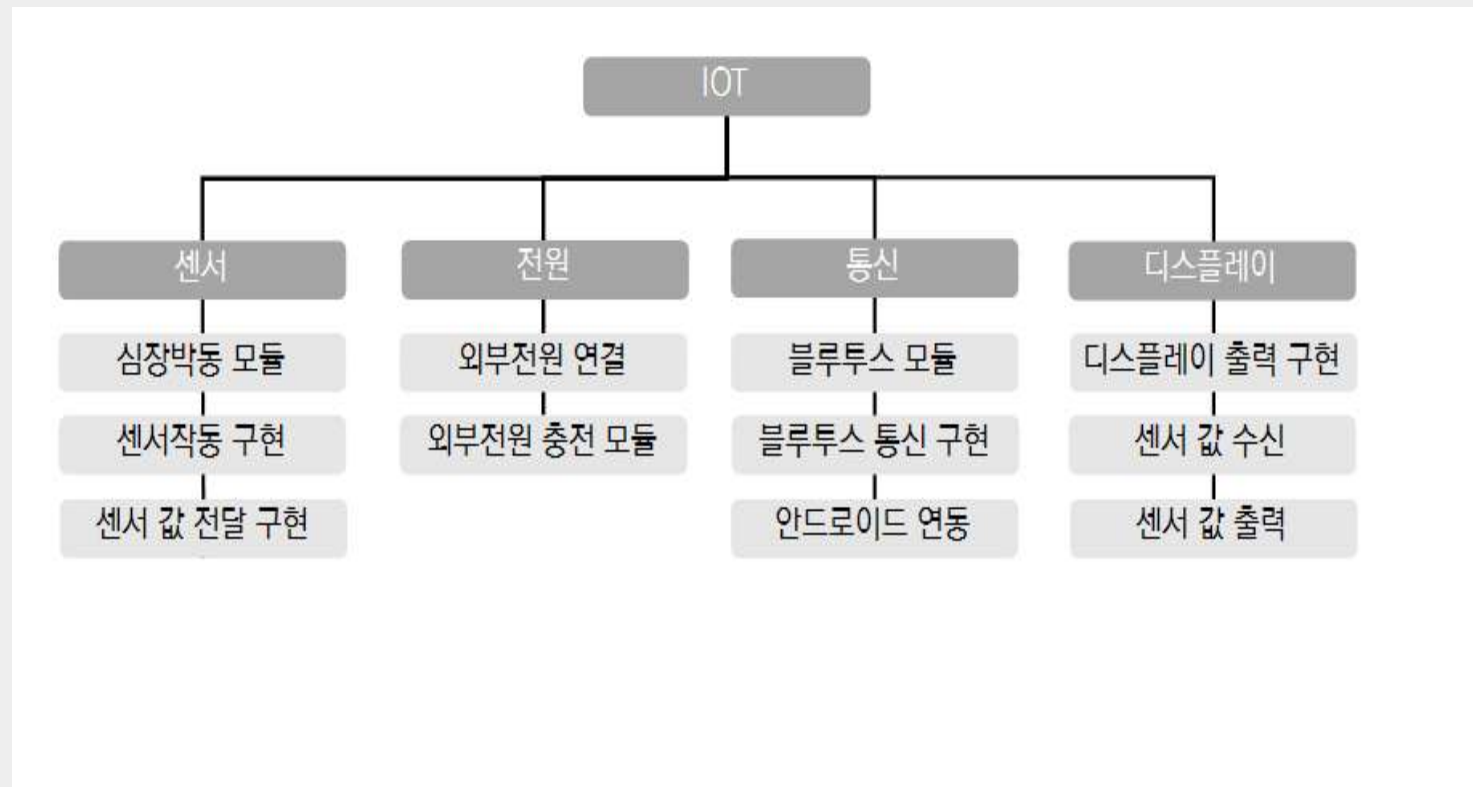
ARDUINO

### 03. IOT 개발

#### 1) 작업분할구조도

## 03 IOT 개발

### 1. 작업분할구조도



### 03. IOT 개발

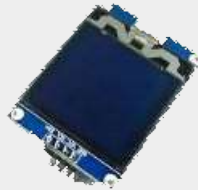
#### 2) 주요기술

## 03 IoT 개발

### 2. 주요기술



아두이노 프로 미니



OLED 디스플레이

심박수와 산소  
포화도 등 각종  
정보를 출력할  
디스플레이



HC-06 블루투스 모듈

앱과 웨어러블  
기기 사이의  
통신을  
담당하는 모듈



리튬 폴리머 배터리

기기의  
전원공급을  
위한 배터리



심장박동 모듈

심박수와 산소  
포화도를 측정할  
수 있는 모듈



END OF DOCUMENT

THANK YOU