



2 0 1 8 1 7 8 1 최 성 호



# CONTENTS

## 01. 소프트웨어 공학

---

소프트웨어  
소프트웨어 공학  
소프트웨어 프로세스

## 02. 깃허브

---

깃허브  
깃허브 용어 정리

## 03. 데이터베이스

---

데이터베이스  
파이어베이스



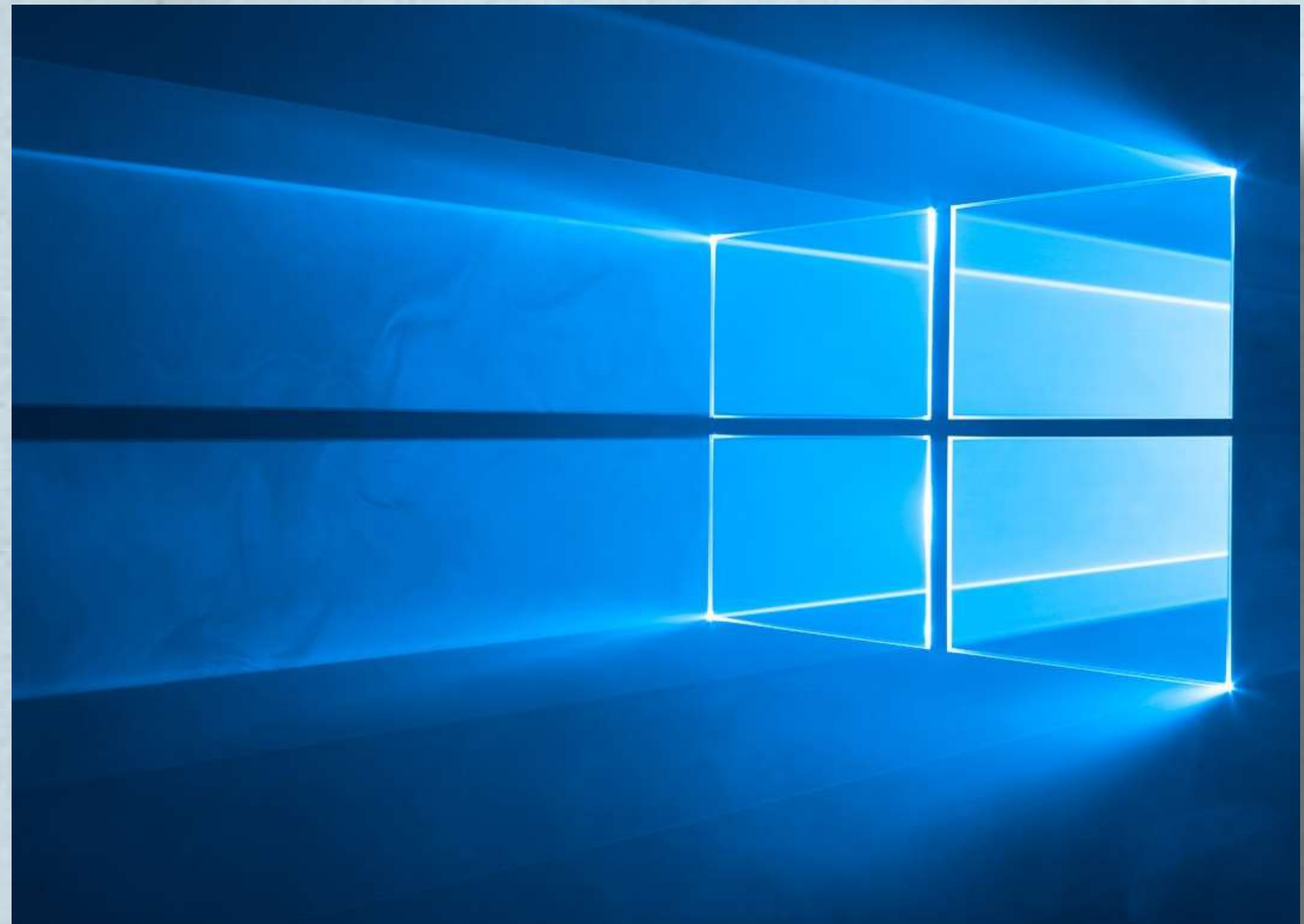
# 01

## 소프트웨어

---

프로그램과 달리 코드뿐만 아니라 자료구조,DB구조 등 모든 산출물의 지칭하는 말로 크게 시스템 S/W , 응용 S/W가 있다.

대표적인 시스템 S/W은 운영체제(OS)  
응용 S/W는 워드프로세서 , DBMS 등이 있다.





## 1-1. 소프트웨어

### S/W의 특징

---

#### 1. 제조가 아닌 개발

개인능력 별 결과물 차이가 큼

#### 2. 소모가 아닌 품질저하

H/W에 비해 오래 사용하면 닳지 않고 고장 발생빈도 낮고 기능 동일

- 개발과정이 복잡하고 많은 인력이 참여하고 개발 기간이 길다 -> S/W 공학의 존재 이유

### S/W,H/W 실패 곡선

---

#### 1. H/W 실패 곡선

초기 실패율이 높고 점점 낮아지다  
환경문제로 다시 올라감  
(욕조 모양의 곡선)

#### 2. S/W 실패 곡선

이상적인 곡선은 초기 실패율이 높지만 수정으로 인해 점점 낮아지고 다시 높아지진 않음

하지만 실제 곡선은 변경점으로 인해 다시 높아짐

### S/W 당면 과제

---

#### 1. H/W의 비해 S/W 개발의 느린 발전속도

#### 2. 새로운 S/W의 대한 사용자 요구 증가

\*H/W와 S/W의 개발 방법의 근본적인 차이 때문\*

#### 3. S/W도 관리가 필요함 비용-일정-개발자



## 1-2. S/W공학

### S/W 공학

---

### S/W + 공학

소프트웨어 개발 과정에 공학적인 원리를 적용

소프트웨어를 개발하는 것을 의미함

### S/W 의 목적

1. S/W 개발의 어려움 해결
2. 효율적 개발을 통한 생산성 향상
3. 고품질 소프트웨어 제품 생산 - 사용자의 만족



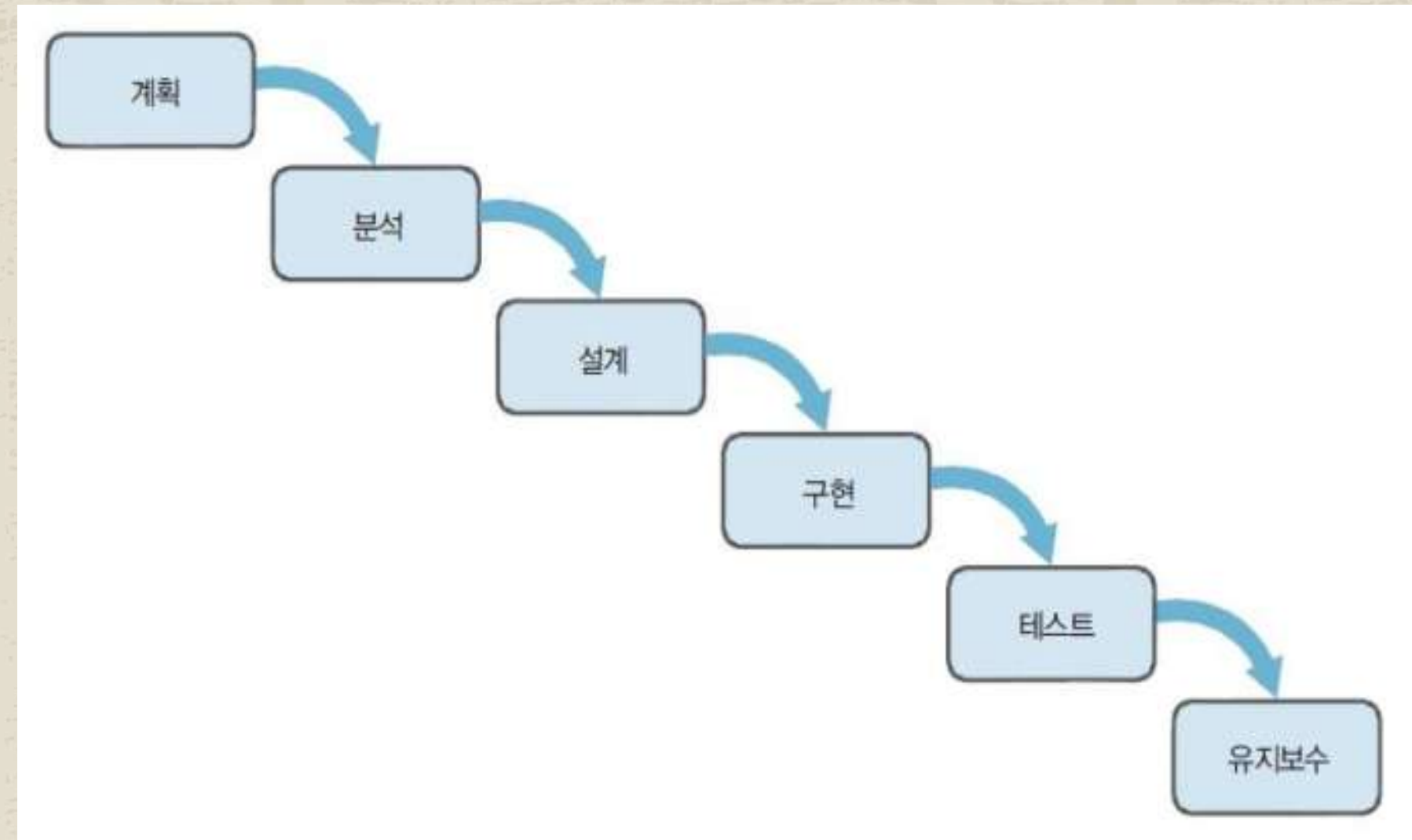


## 1-2. S/W공학

### S/W 공학 개발과정 (SDLC)

---

S/W 공학의 개발 하는 과정으로  
계획 단계부터 유지보수 단계  
까지의 흐름





## 1-2. S/W공학

### 1. 계획

기능점수 모델과 COCOMO 모델등을 이용하여  
비용 산정, 작업분할 구조도(WBS)를 이용하여 프로젝트를 작은 단위로 나눈다.

### 2. 요구 분석

기존 시스템의 문제점 파악 - 새로운 요구사항  
도출- 다이어그램 작성

개발 방법론에 따른 표현 도구

구조적 방법론 - 정보공학 방법론 - 객체지향  
방법론

최종 산출물 : 요구 분석 명세서

### 3. 설계

설계 원리 : 분할과 정복, 추상화, 단계적 분  
해, 모듈화, 정보은닉

아키텍처 스타일 , 객체지향 설계 , GOF의  
디자인 패턴

모듈 평가 기준 : 응집도, 결합도



## 1-2. S/W공학

### 4. 구현

코드 작성을 통한 프로그래밍  
표준 코딩 규칙 엄수

### 5. 테스트

개발자 또는 사용자 시각에 따른 분류

사용되는 목적에 따른 분류

품질 특성에 따른 분류

소프트웨어 개발 단계의 따른 분류

### 6. 유지 보수

수정 유지보수

적응 유지보수

기능보강 유지보수

예방 유지보수



## 1-3. S/W프로세스

### 프로세스

---

일을 처리하는 과정 또는 순서

#### 목적

1. 이전의 얻은 노하우 전달
  2. 시행착오 감소
- \*가이드 역할\*

### S/W 프로세스 모델

---

S/W를 개발하기위해 SDLC를  
체계화한 개념  
순차적인 단계로 이루어짐

#### 목적

고품질의 S/W 생산



## 1-3. S/W프로세스

### 선형 순차적 모델 (폭포수 모델)

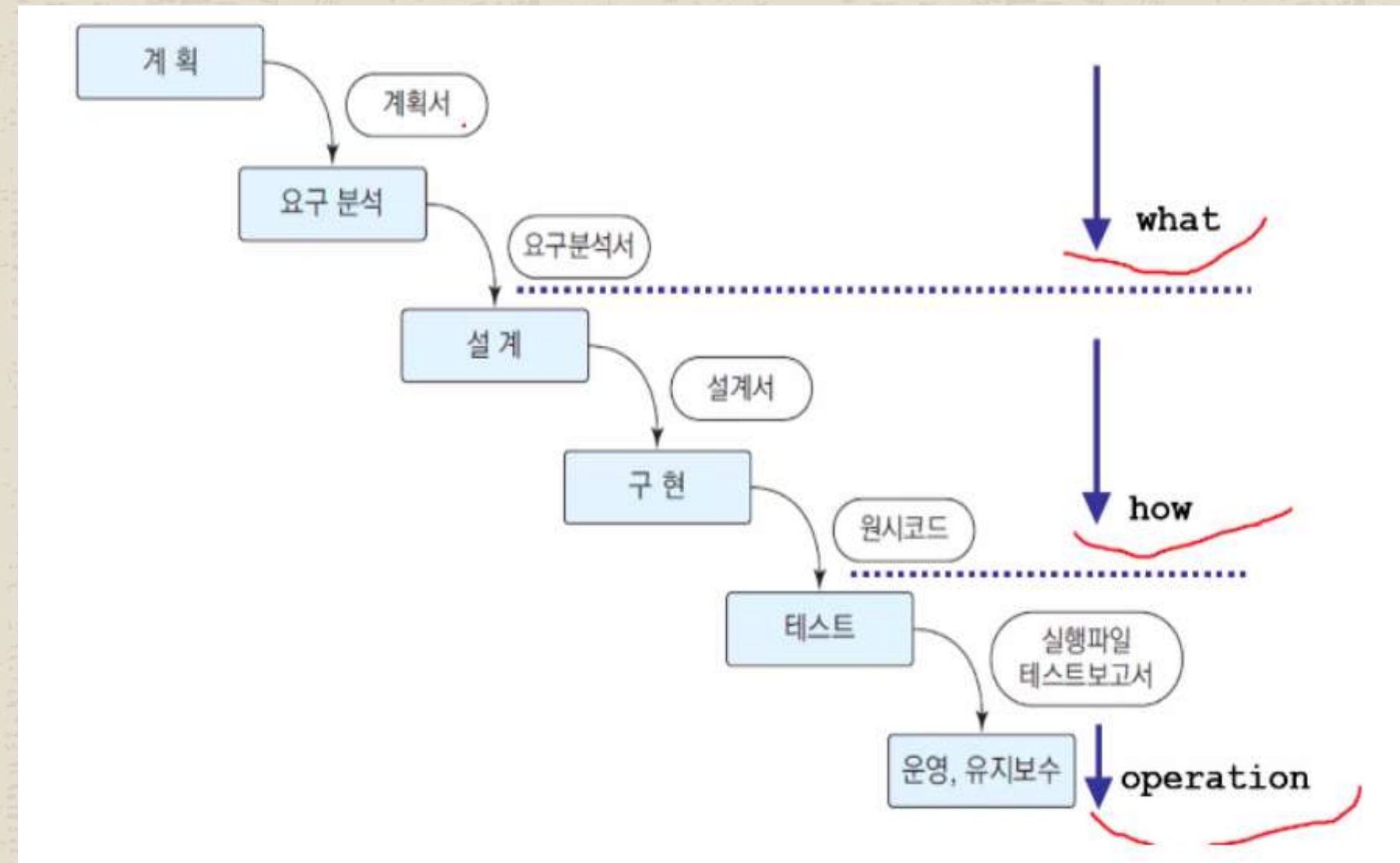
폭포처럼 차례대로 내려오는  
모델

장점

관리 용이  
체계적인 문서화  
변화가 없는 프로젝트에 적합

단점

선행 단계가 수행되어야 다음단계 수행  
가능  
사용자가 중간에 대략적인 내용 확인  
불가



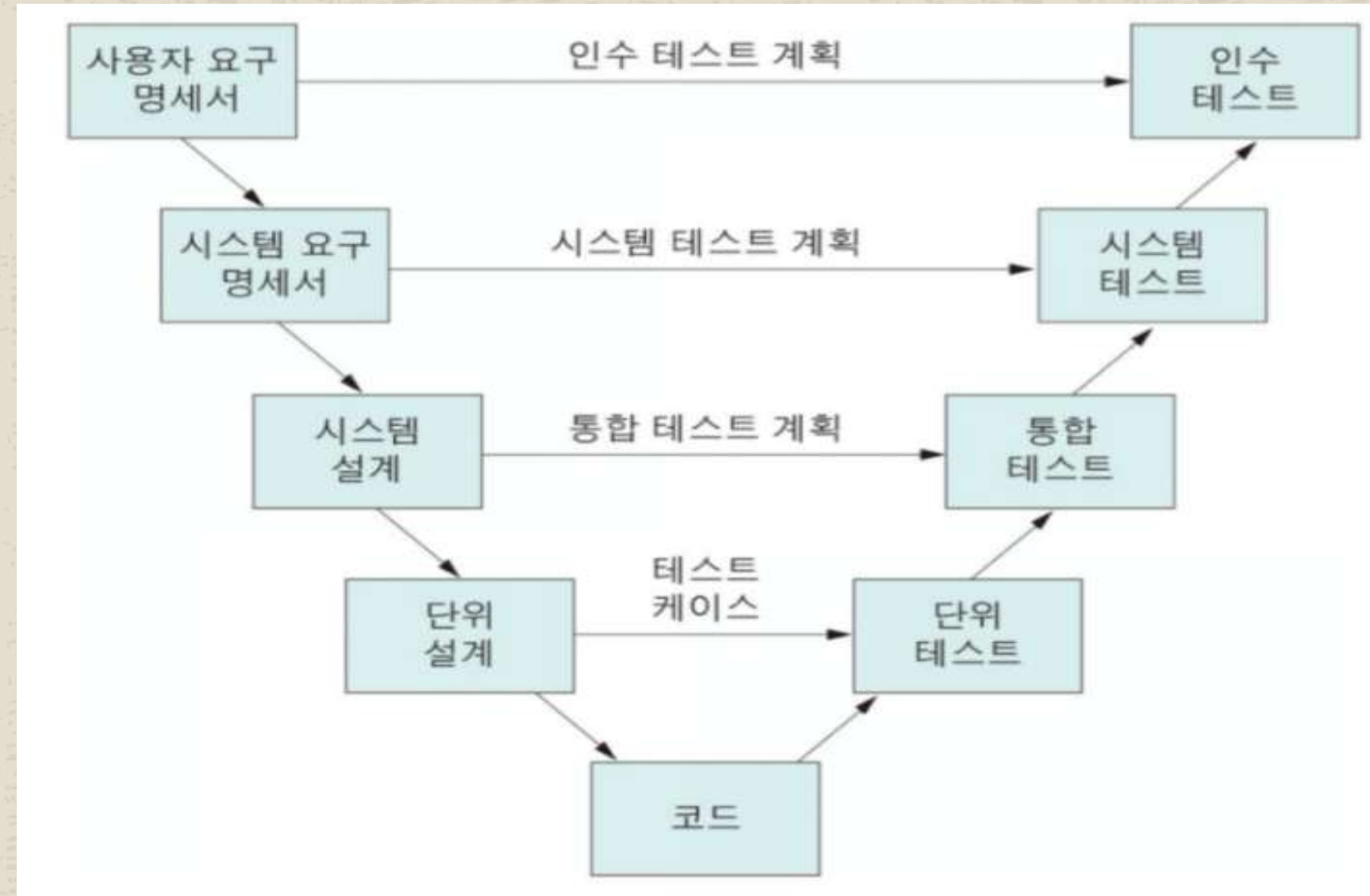


## 1-3. S/W프로세스

### V 모델

폭포수모델에 테스트 단계를 확장 시킨  
모델

산출물 중심인 폭포수에 비해  
테스트 단계가 중심인 모델





## 1-3. S/W프로세스

### 진화적 프로세스 모델 (프로토 타입)

---

#### 프로토 타입

생산전에 시험차 만들어 보는 시제품  
완전한 절차가 아닌 사용자의 요구를 받아  
일단 만들고 사용자의 피드백을 수용하기 위  
한 도구

개발 절차를 통해 만들고 사용자들에게 보여  
줌

### 프로토 타입의 개발 절차

---

#### 1. 요구분석

1차 간략히 정리후 2,3,n차를 통한 최종 프  
로토 타입 개발

#### 2. 설계

사용자와 의사소통의 도구 정도로 설계  
입출력 화면을 통한 사용자 중심 인터페이스  
설계

#### 3. 개발

입력 화면을 통한 사용자의 요구항목 확인  
출력 화면을 통한 사용자의 만족도 확인



# 02

깃허브

---

버전 관리를 위한 서버 저장소 및 프로젝트  
개발을 위한 협업 관리 서비스





## 2-1. 깃허브의 정의

### 버전 관리

---

시간흐름에 따라 파일 또는 폴더에 대한  
변경 사항을 관리

### 기능

#### 1. 버전의 저장과 백업

변경사항의 자세한 확인

#### 2. 여러 사용자에게 버전 이력 관리

파일이 언제 어떻게 추가,삭제됐는지 확인  
가능

### 깃허브

---

버전 관리를 위한 서버 저장소 및  
프로젝트 개발을 위한  
협업 관리 서비스

분산 버전 제어 S/W(깃)  
을 위한 웹 저장소 + 커뮤니티

깃허브를 사용하면  
오픈소스 기반 프로젝트에  
누구나 참여할수 있음



## 2-1. 깃허브의 정의

### 깃허브의 장점

---

1. 무료
2. 기존 오픈소스 프로젝트는 권한제어, 해당 커뮤니티 가입등 걸리는점이 있는데 오직 깃허브 아이디 하나로 모든게 해결됨
3. fork 기능 하나로 내 저장소에 손쉽게 저장하고 오픈소스의 수정 및 이슈 제기가 가능해짐

### 분산 버전 관리 시스템

---

각 개발자가 중앙 서버에 접속되지 않아도 코드작업이 가능하며 독립적인 결과물을 병합,거절 할 수 있도록 버전을 관리해주는 시스템

각각의 프로젝트의 로컬저장소와 원격저장소를 분리하여 독립적인 코드 관리가 가능



## 2-2. 깃허브 용어 정리

**fork**

다른 사람의 저장소에 있는  
프로젝트를 내 공간으로 복사하는 것

---

**upstream**

깃허브에 있는 타인의 저장소

---

**origin**

깃허브에 있는 내 저장소

---

**local**

내 컴퓨터에 있는 저장소

---

**pull**

로컬 저장소에서 깃허브로 옮기는것

---

**push**

깃허브에서 로컬저장소로 옮기는것

---

**fetch/merge**

저장소의 내용을 수정했을때 다른사람  
의 내용을 볼수있게 최신화하는것

**clone**

깃허브의 저장소내용을 바로 내 로컬  
저장소에 옮기는것

**pull request**

저장소 소유자에게 변경사항을 확인하  
고  
원하면 반영하라는 뜻

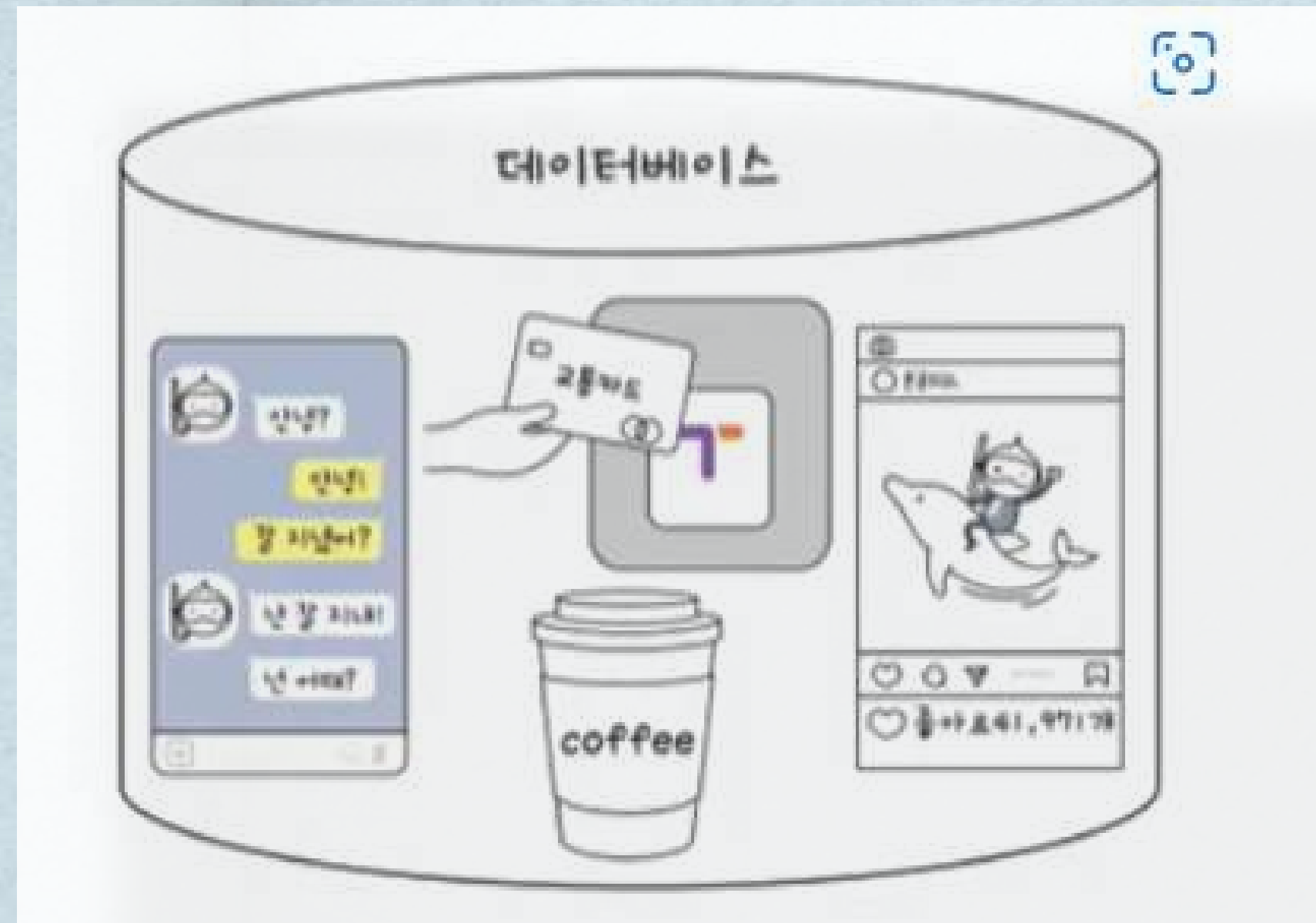


# 03

## 데이터베이스

여러 사람이 사용할 목적으로  
데이터를 보관하는 저장소

팀프로젝트 DB담당을 맡아 공부하게 됨





### 3-1. 데이터베이스

#### 데이터베이스

---

데이터베이스란  
쉽게 말해 정보들의 저장소이며  
DBMS라는 관리 시스템으로  
관리함

#### DBMS

---

데이터베이스를 관리하며 응용 프  
로그램들이 데이터베이스를 공유하  
며 사용할수 있는 환경을 제공하는  
S/W

#### 관계형 데이터베이스

---

대표적인 데이터베이스로  
2개 이상의 데이터베이스와  
테이블등을 연결하기 위해  
고유한 식별자를 사용하는  
데이터베이스  
테이블 형태로 이루어져 있으며 정  
보를 요청할때 질의문을 쓴다.



### 3-1. 데이터베이스

뷰	질의문(query)	데이터베이스 관리자 (DBA)
사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된 가상 테이블	해당하는 데이터베이스의 정보를 보기위해 사용자가 DBMS에게 보내는 요청	한 조직내에서 데이터베이스에 관련된 모든것(설치,구성,감시,관리) 등의 총 책임자



### 3-1. 데이터베이스

#### DDL

데이터베이스를 정의하는 언어를  
말하며, 데이터를 생성하거나  
수정하거나 삭제하는 등  
데이터의 전체 골격을 결정하는  
역할의 언어

create  
alter  
drop  
truncate

#### DML

정의된 데이터베이스에 입력된 레  
코드를 조회하거나 수정하거나  
삭제하는 등의 역할을 하는 언어

select  
update  
insert  
delete

#### DCL

데이터베이스에 접근하거나 객체에  
권한을 주는 등의 역할을 하는 언어

grant  
rovoke  
commit  
rollback



## 3-2. 파이어베이스

### 파이어베이스

---

구글에서 만든 모바일 플랫폼  
서버가 해야하는 일을 미리 만들어  
놓아 사용자가 서버 개발 없이  
서버를 사용할 수 있게 해주는  
플랫폼

Firestore는 Android Studio를  
사용하여 원하는 기능에 대한  
부분의 SDK를 설치한 뒤에  
import를 통해 쉽게  
사용이 가능하다.

현재 구현을 위해 공부중





## 3-2. 파이어베이스

### Realtime database

---

파이어베이스의 DB

Firestore에 비해 구형  
검색 기능 쿼리가 상당히 부족해  
요즘 Firestore로 넘어가는 추세

### Firestore

---

파이어베이스의 DB

Firestore는 기존 RealtimeDatabase보다 쿼리, 보안, 색인 기능이  
강력해지긴 했지만 상용  
데이터베이스인 Oracle, MySQL  
보다는 아직 약함



## 3-2. 파이어베이스

### 안드로이드 스튜디오와 파이어베이스 연동하는법

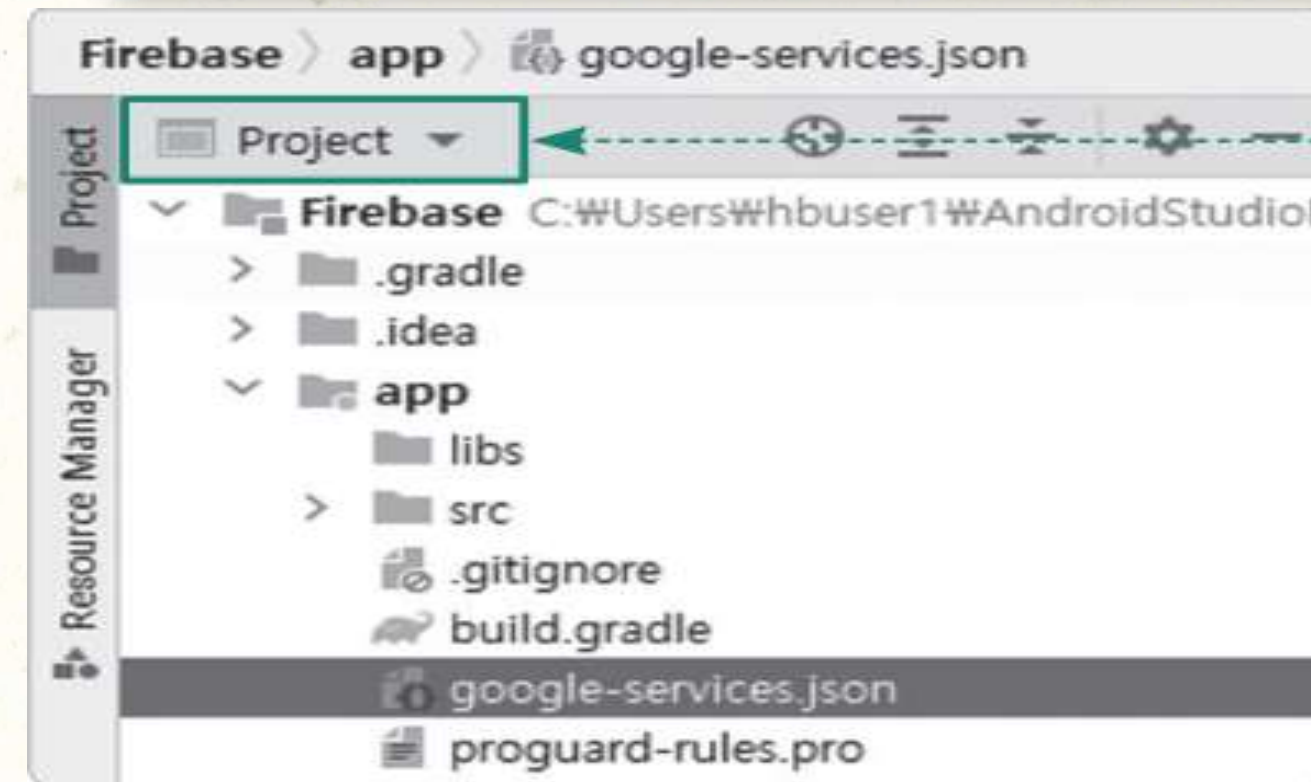
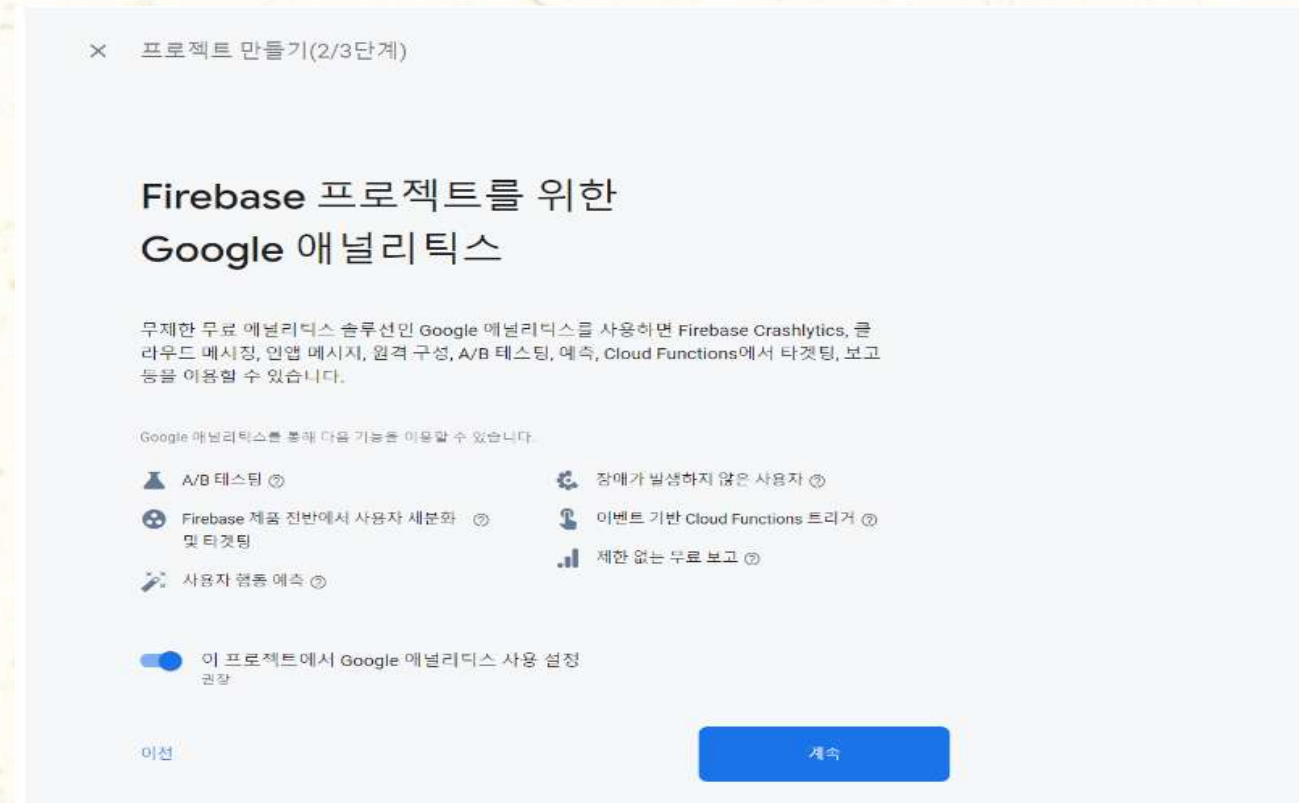
---

1.  
파이어베이스 홈페이지에서  
구글 애널리틱스 프로젝트 구성

2.  
안드로이드 스튜디오에  
파이어베이스라는 이름으로  
프로젝트 생성

3.  
파이어베이스 홈페이지에 제공되는  
json파일과SDK파일 삽입

4.  
연동 확인후  
제공되는 기능 사용





THANK

YOU

2 0 1 8 1 7 8 1

최 성 호