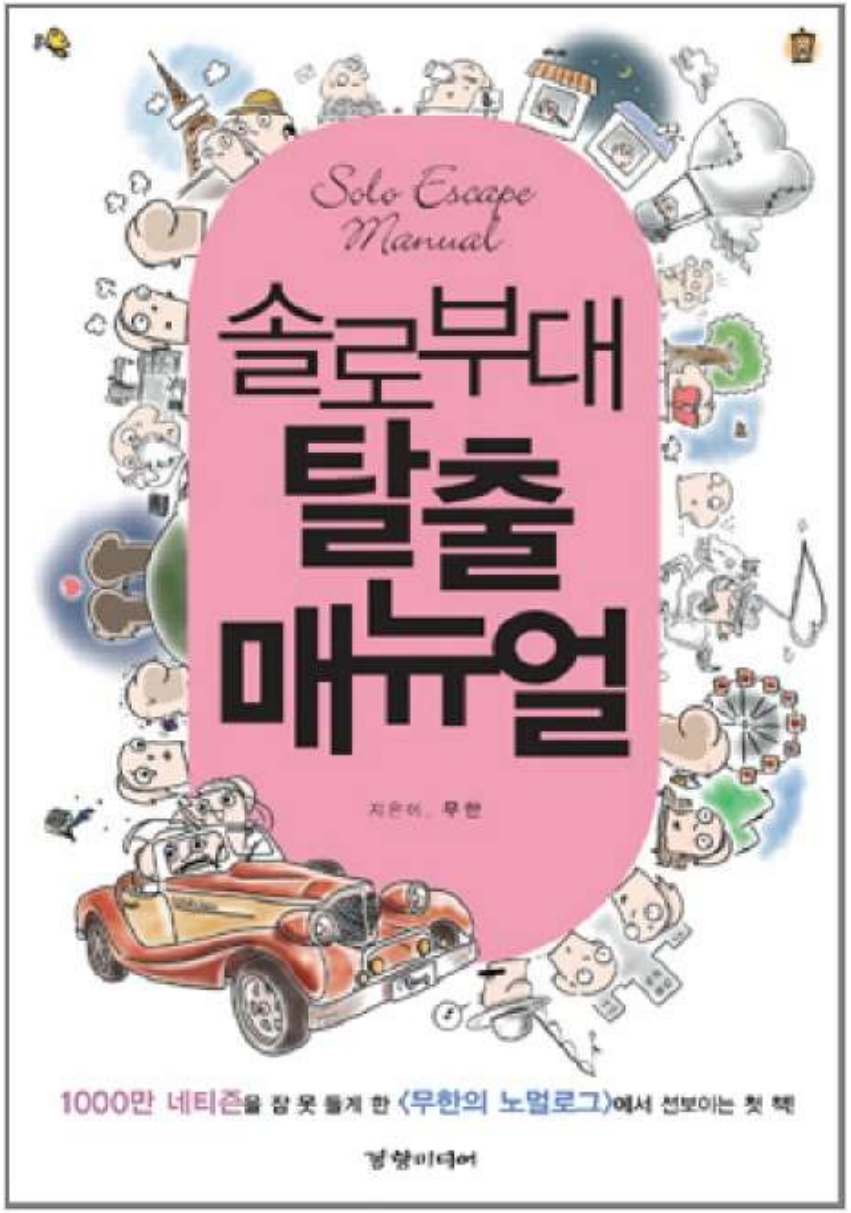
소프트웨어 공부하는 법

이민석
소프트웨어학부

KOOKMIN UNIVERSITY

오늘의 이야기 순서

- 소프트웨어나 그냥 공부나..
- 소프트웨어 개발자의 성장 단계 (aka. 왜 어렵나?)
- 소프트웨어 배우는 법
- 소프트웨어 학부 / 연계전공
- 학교 밖에서의 소프트웨어 학습
- 결국 배워야 하는 것


Solo Escape Manual
솔로부대
탈출
매뉴얼
지은이. 무한
1000만 네티즌을 잠 못 들게 한 〈무한의 노멀로그〉에서 선보이는 첫 책
김영사

ASKY

코딩이나 그냥 공부나...

책을 따라 진도를 나가거나

모든 수업에서는 문제가 제시된다.

남이 만든 문제는 **공부하면** 결국 풀 수 있게 된다.

그러나 문제를 풀기만 하면,

문제 낸 사람을 능가할 방법이 없다.

최고가 되려면...

문제(why와 how를 적용할 대상)을 발굴하고

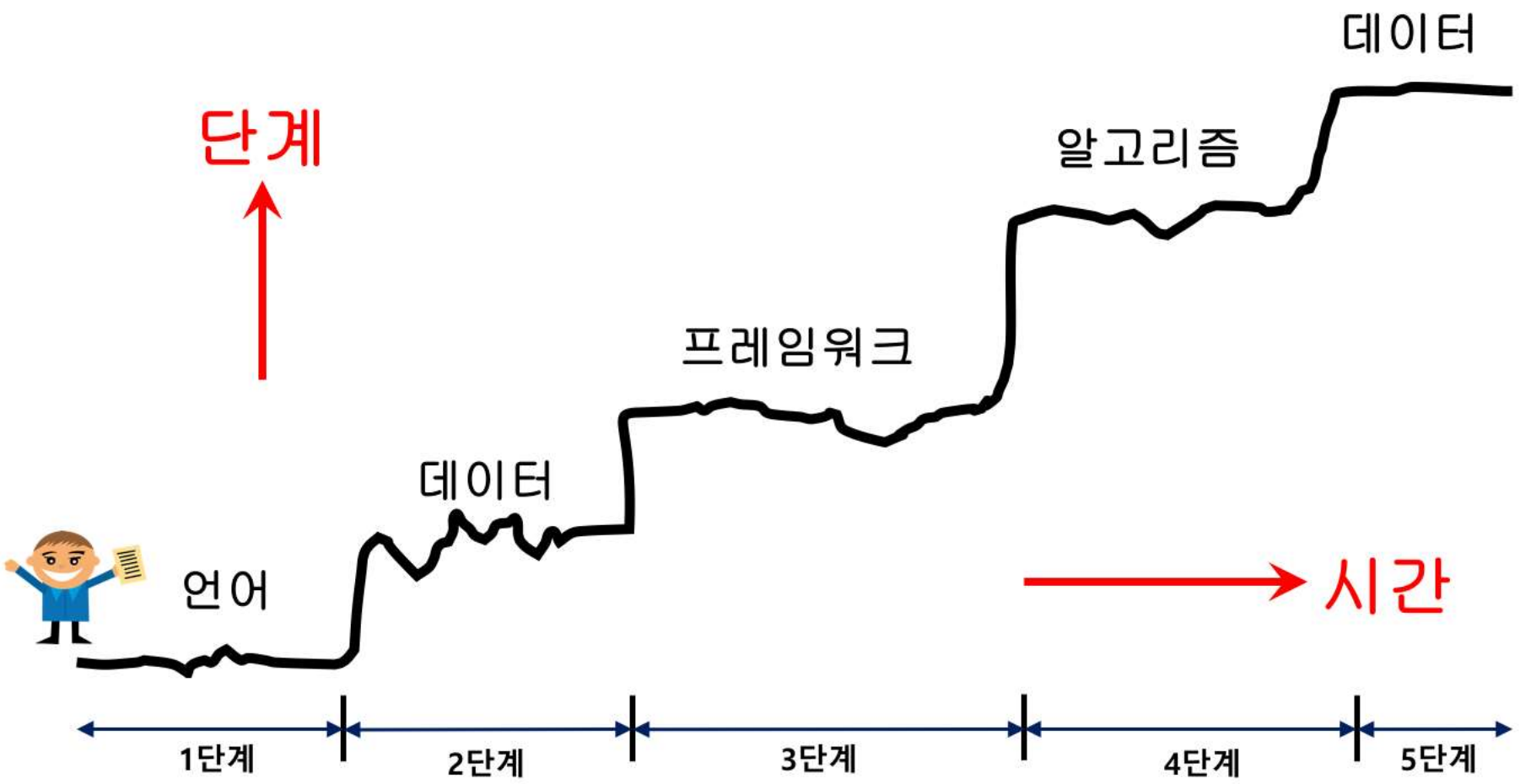
내가 만든 문제를 풀면서 배워야 한다.

경험 주도의 학습이 필요

- 작은 성취를 통해 어느 틈에 뭔가 할 수 있게 되는 학습

https://www.youtube.com/watch?v=KgzLzbd-zT4

개발자 성장 단계



http://hl1itj.tistory.com/136

소프트웨어를 배우는 방법 (1/2)

1. 책을 사기 전에 무엇을 만들까를 먼저 생각한다.
   - 전혀 새로운 것일 필요가 없다. 내가 자주쓰는 걸 도전하자
2. Youtube를 보면서 Hello World 따라해본다.
   - 유투브에서 "언어-이름 hello world" 를 검색
3. 짧은 동영상 강의를 영혼 없이 본다.
   - Youtube, InfLearn, 생활코딩, 구글… "언어-이름 tutorial (튜토리얼)"
4. 책을 사서 아주 빠르게, 빛의 속도로 읽는다.

5. [1]에서 만드려고 했던 것의 최소한을 만든다.
   - 예) 첫 날은 그냥 그림을 그려서 되는 것처럼 보이게 한다.
   - 시작이 반이다.

소프트웨어를 배우는 방법 (2/2)

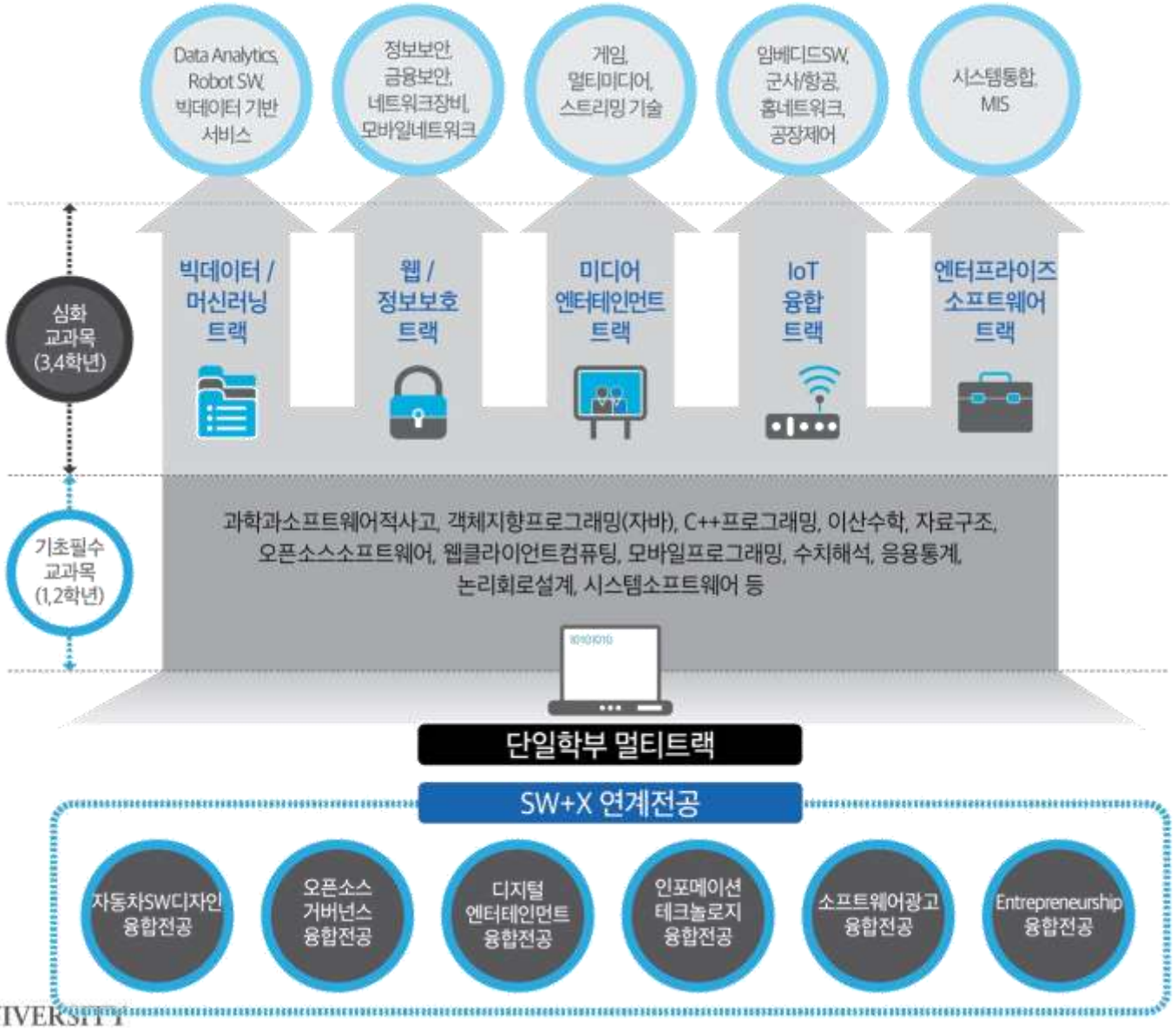
6. 만들려다가 모르는 부분의 동영상을 자세히 본다.
7. 만들려다가 모르는 부분의 책을 차분히 본다.
8. 그래도 안되는 부분은 멘토를 찾는다.
9. 완성되면
   친구에게, 애인에게, 부모님에게, 조카에게 자랑을 한다.
   - 이 부분이 핵심이다.

   ➔ 자랑하고 피드백 받고 수정해야 실력이 는다.

10. 최소한을 조금 늘려간다. ➔ goto [6] 반복

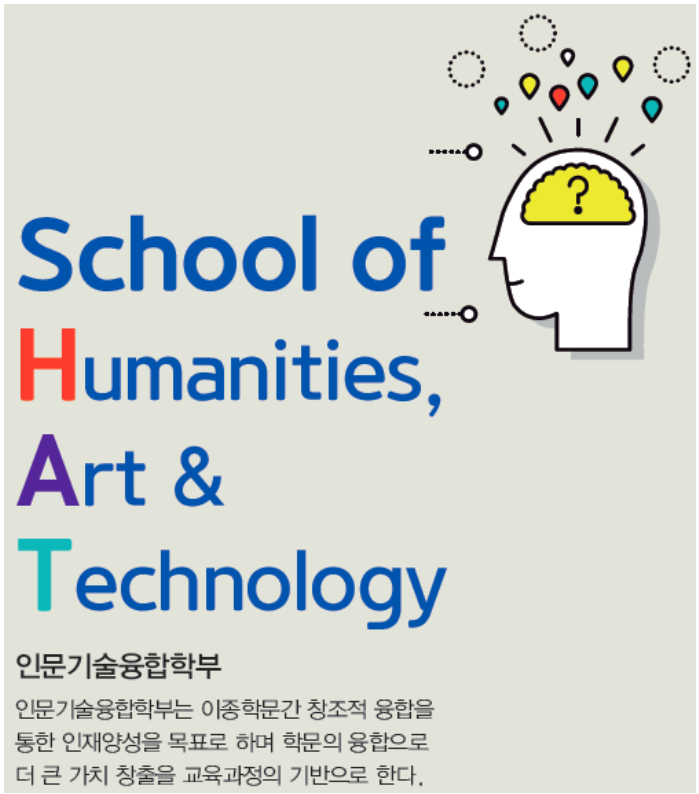
11. 그러다보면 어느날 자기도 모르게 계단을 올라선다.
12. 뭔가 느낌이 있을 때, 소프트웨어 전공과목 공부를 한다.

KMU 소프트웨어 학부의 커리큘럼



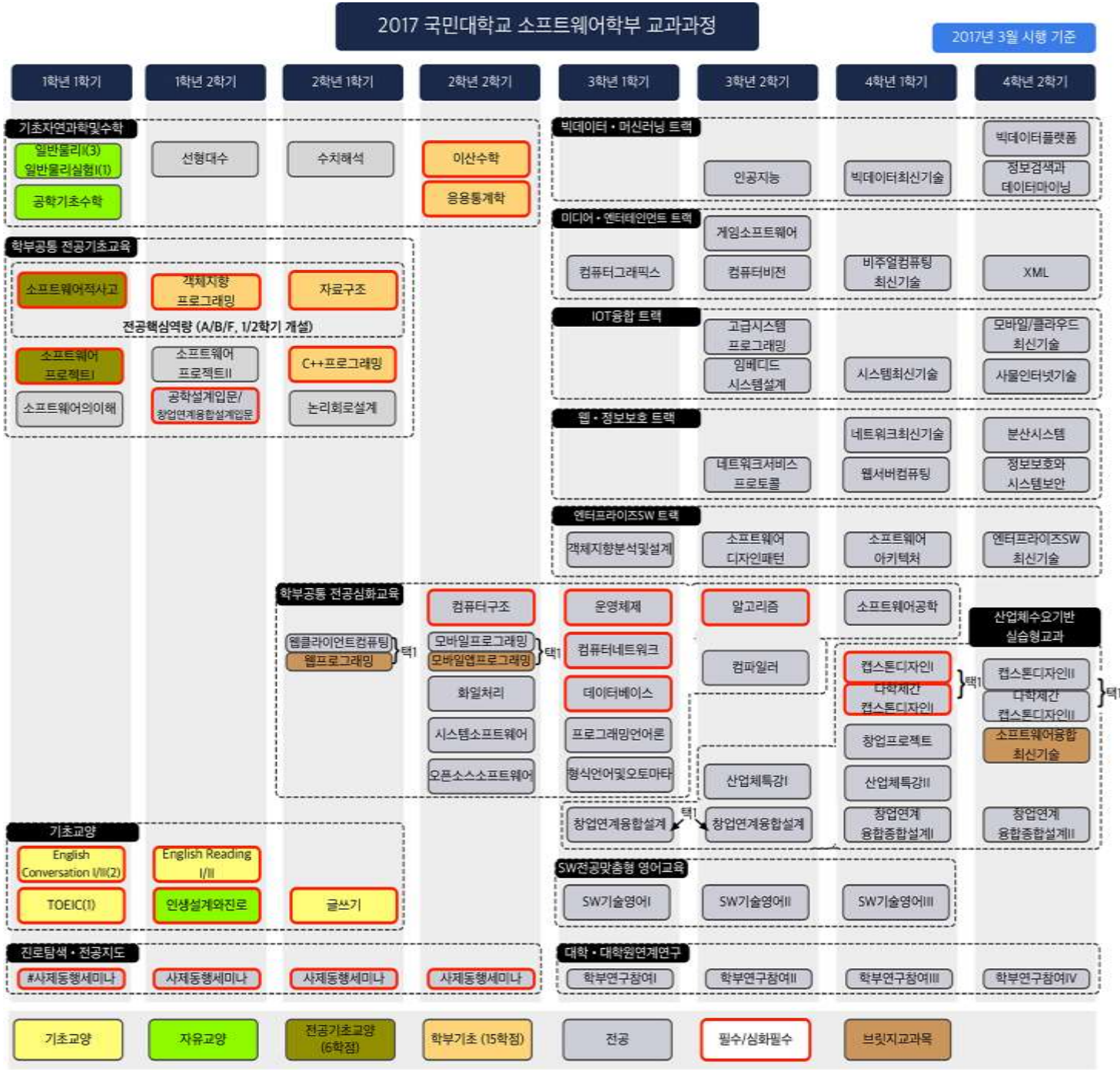
소프트웨어 학부 참여, 융합 / 연계 전공

- 자동차 소프트웨어 디자인 융합
  - 전공자동차+소프트웨어+자동차운송디자인
- Entrepreneurship 융합전공
  - 창업지원단+공업디자인+경영학+경영정보+교양+소프트웨어
- 오픈소스 거버넌스 융합전공
  - 소프트웨어+행정정책+사법
- 소프트웨어 광고 융합전공
  - 소프트웨어+광고홍보학
- 인포메이션 테크놀로지전공
  - 소프트웨어+경영정보
- 디지털 엔터테인먼트 전공
  - 소프트웨어+영상디자인
- 바이오4차산업거버넌스연계전공
  - 소프트웨어+바이오발효융합+행정정책
- 소프트웨어 미디어 전공
  - 소프트웨어+미디어학
- 자기설계융합전공



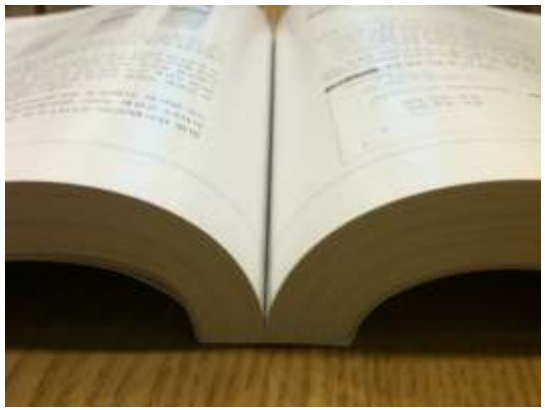
그런데, 쉽지가 않아요

AB/F? 들어는 봤나?



소프트웨어 학습 Resource

- 뭐니뭐니 해도 구글 검색
  - "~~ 어떻게 하나요?", "how to ~~", "~~ 따라하기", "~~ tutorial", …
- Online resource
  - https://programmers.co.kr/
  - https://www.inflearn.com/ (인프런)*
  - https://www.codeonweb.com/
  - https://opentutorials.org/ (생활코딩)
  - https://edu.goorm.io/ (구름에듀)
  - … kmooc … mooc …
- 온라인 질의 응답
  - http://hashcode.co.kr/
  - https://stackoverflow.com/
- Offline
  - 기초부터 빡쎄게 하는 곳들이 좀 있기는 하지만..



인프런 추천 강의

- 파이썬을 이용한 프로그래밍 입문
  - 데이터 과학을 위한 파이썬 프로그래밍
    - https://www.inflearn.com/course/python-파이썬-입문-강좌/
  - Python Django 프레임워크로 웹서비스 개발하기
    - https://www.inflearn.com/course/dj1ngo-파이썬-장고-강좌/
- 프로그래밍 입문부터 웹-앱 풀스택 배우기
  - 웹-앱 풀스택개발 Boot Camp 입문부터 서비스 개발까지
    - https://www.inflearn.com/course/풀스택개발-full-stack/
- 비전공자를 위한 R을 이용한 데이터 자동화
  - R 문법 기초
    - https://www.inflearn.com/course/r-기초-데이터-분석/
  - R 데이터 시각화
    - https://www.inflearn.com/course/r-시각화-기초/
- iOS 1인 개발자 되기 전체 과정
    - https://www.inflearn.com/learningpath/ios-개발자-되기/
- JAVA - JSP - SPRING 비전공자 자바개발자로 취업하기
    - https://www.inflearn.com/learningpath/java-gibon/

그리고, 수학 = 사고와 변화에 대한 관점 훈련 (1/2)

수학이 심하게 필요한 영역도 있고

적당한 수학이 필요한 영역도 있음

하지만 수학적, 논리적 사고력은

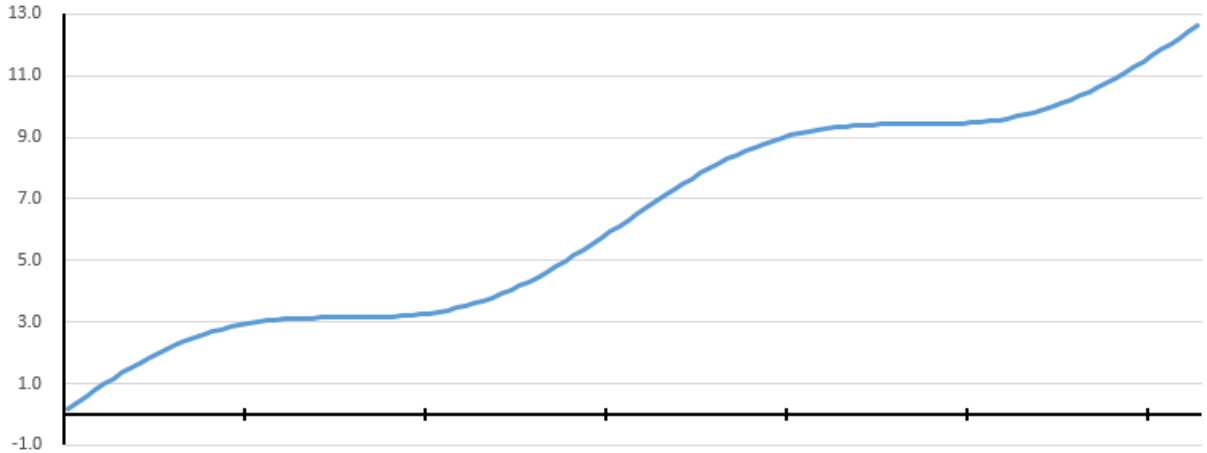
관점이 교차하는 영역에서 소통의 도구로서

변화에 적응하고, 변화를 이끌고 설명하는 강력한 수단임

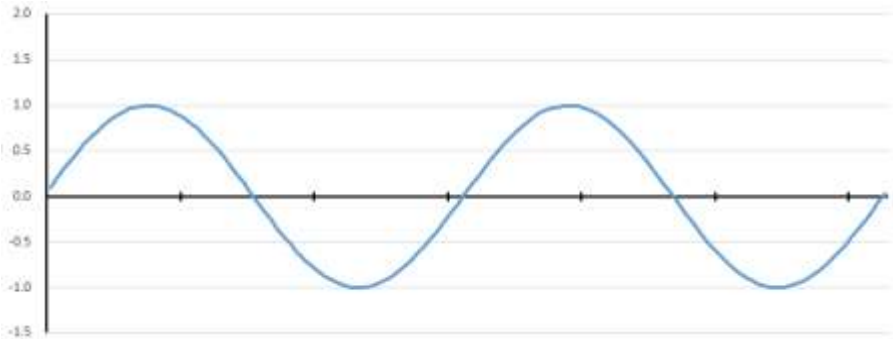
선형대수, 이산수학, 확률 통계

그리고 변화를 이해하고, 설명하기 위한 미적분

수학 = 사고와 변화에 대한 관점 훈련 (2/2)

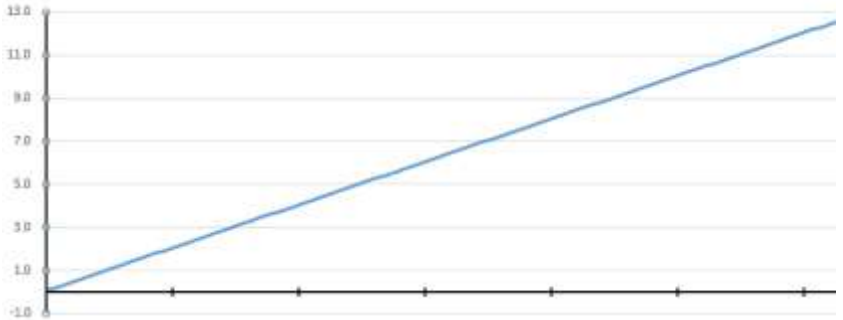


← 세상의
어떤 변화

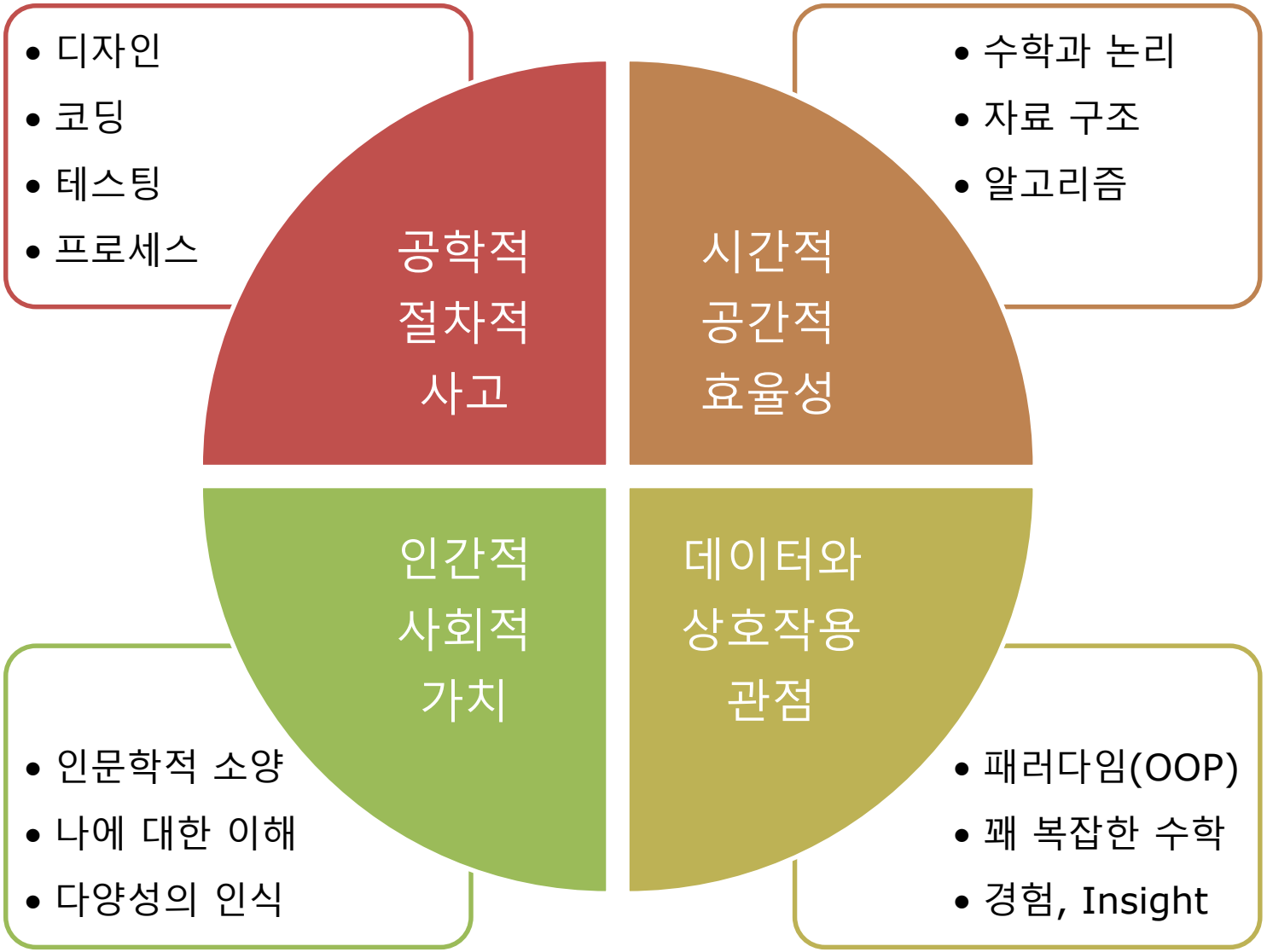


← 내가 이렇게 움직이면

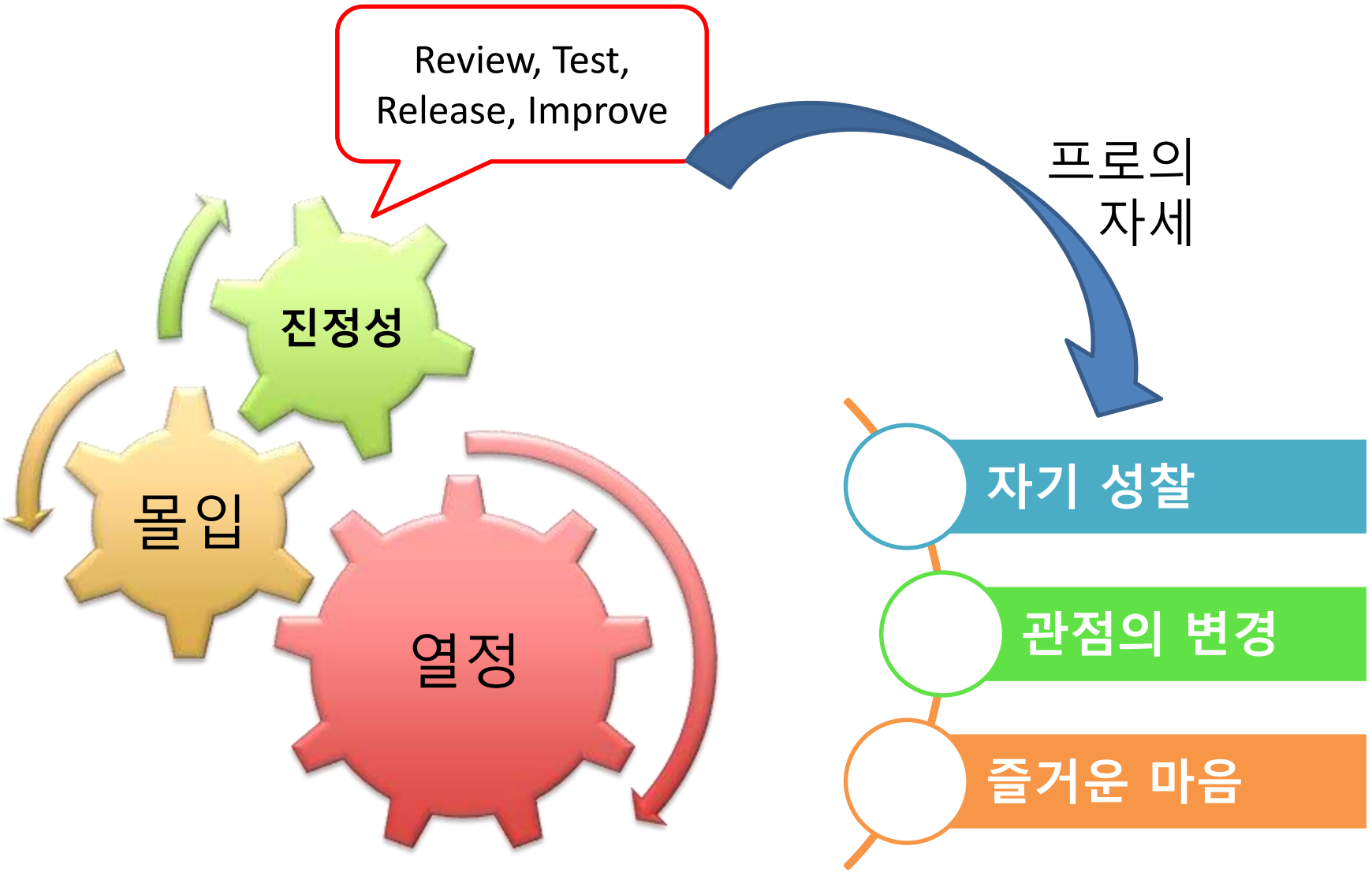
세상은
이렇게 심플해지심 →

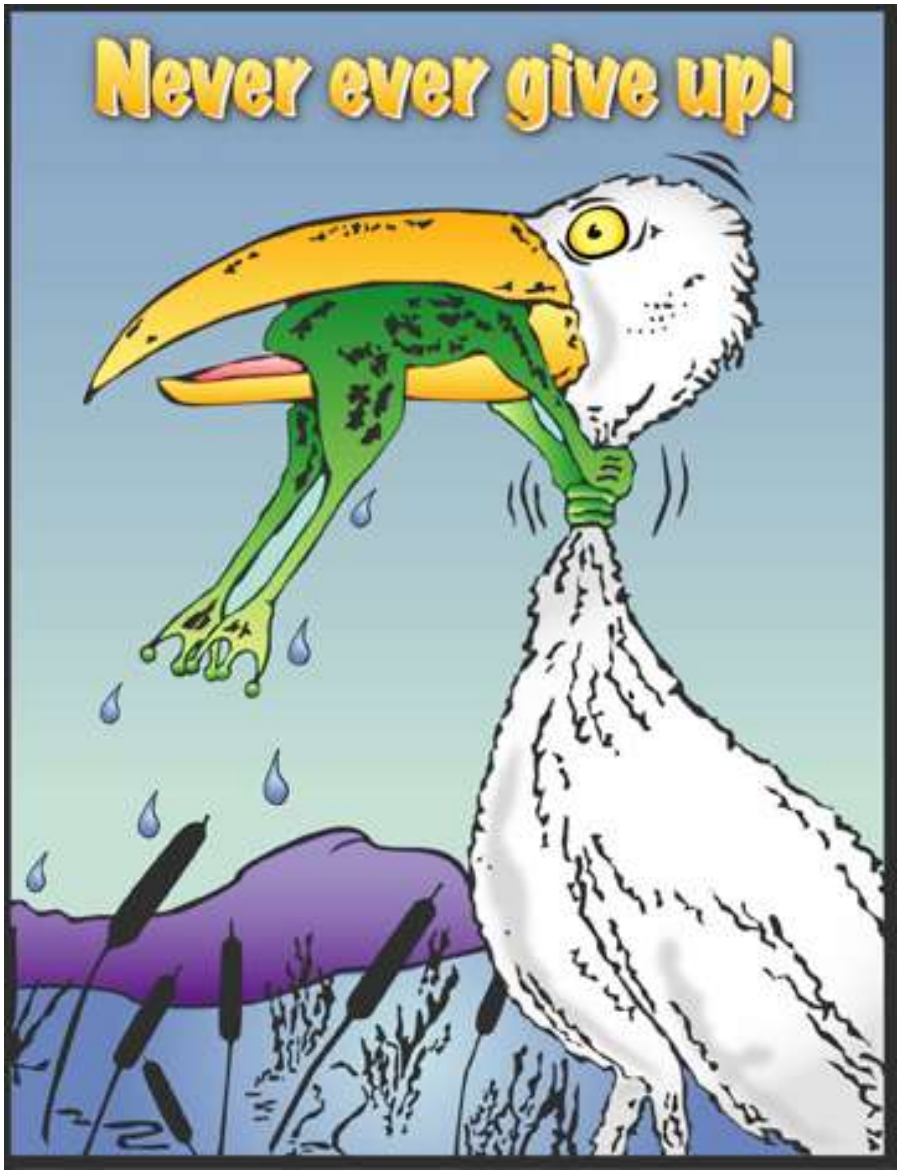


개발자가 되려면, 결국 배워야 하는 것들...



리뷰와 피드백 그리고 지속적 자기 개선




Never ever give up!

Q&A