



# A TO Z PROGRAMMING

A TO Z PROGRAMMING

# PURPOSE

A부터 Z까지 맨땅에 헤딩하기  
우리만의 지식 얻기와 공유하기  
논리적 및 물리적 이해 확장하기

PROJECT

---

계획

분석

설계

구현

테스트

배포

유지보수

# ABOUT CAS

---

CAS는 ‘해봐야 안다’를 모토로, 바텀업 방식의 방향성을 고수합니다.

우리는 직접 경험하고(CONCRETE) 탐구하며(REPRESENTATIONAL) 이해하는(ABSTRACT) 과정을 중요시합니다.

우리는 주어진 개념을 단순히 받아들이는 것을 넘어서, 직접 ‘뜯고, 맛보고, 즐기는’ 과정을 통해 각자의 경험을 공유하고 발전시키고자 합니다. 맨땅에서 시작해 스스로 깨닫는 여정을 기획합니다.

# ABILITY

---

PLANNING



FRONT-END



BACK-END



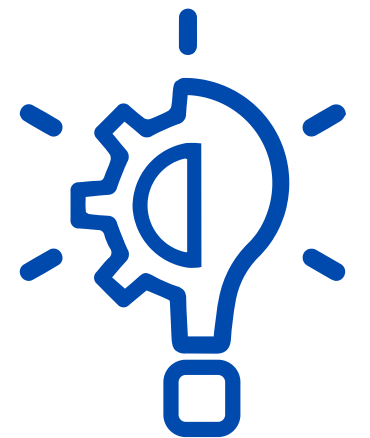
DATABASE



DEPLOYMENT



MONITORING





# PROJECT GUIDELINE

# PLANNING

계획 및 분석

사용 시나리오(기획)

UI/UX 설계(화면 설계, 와이어프레임)

API 스펙 문서화

데이터베이스 모델링

트랜잭션 설계

\*소프트웨어 개발 분야 위주입니다.  
그 외 경영 및 모든 분야는 사이트 참조



# DESIGN&MATIALIZATION

설계 및 구현

시스템 설계(모듈 간 데이터 흐름)

프로그램 설계(각 모듈 내부 알고리즘 및 처리 절차 설계)

백엔드 개발-Spring/SpringBoot

프론트엔드 개발-Vue.js

모듈 개발



# TEST&DEPLOYMENT

테스트 및 배포

단위 및 통합 테스트

E2E 테스트

서버 환경 구성-AWS, GCP, Azure

CI/CD 파이프라인 구성(자동화 배포)-Jenkins

컨테이너 기반 배포-Docker



# MAINTENANCE

유지보수

로그 관리-AWS CloudWatch

지표 모니터링

에러 트래킹-Sentry

운영 이슈 대응







# PROJECT BOTTOM-UP

# BUILD AND FIX MODEL

주먹구구식 모델

바텀업 러닝을 위해서 주먹구구식으로 개발할 필요성이 있습니다.  
해봐야 소프트웨어 개발 프로세스의 이유도 알 수 있기 때문이죠.

*BUILD AND FIX*

*WATERFALL*

*V*

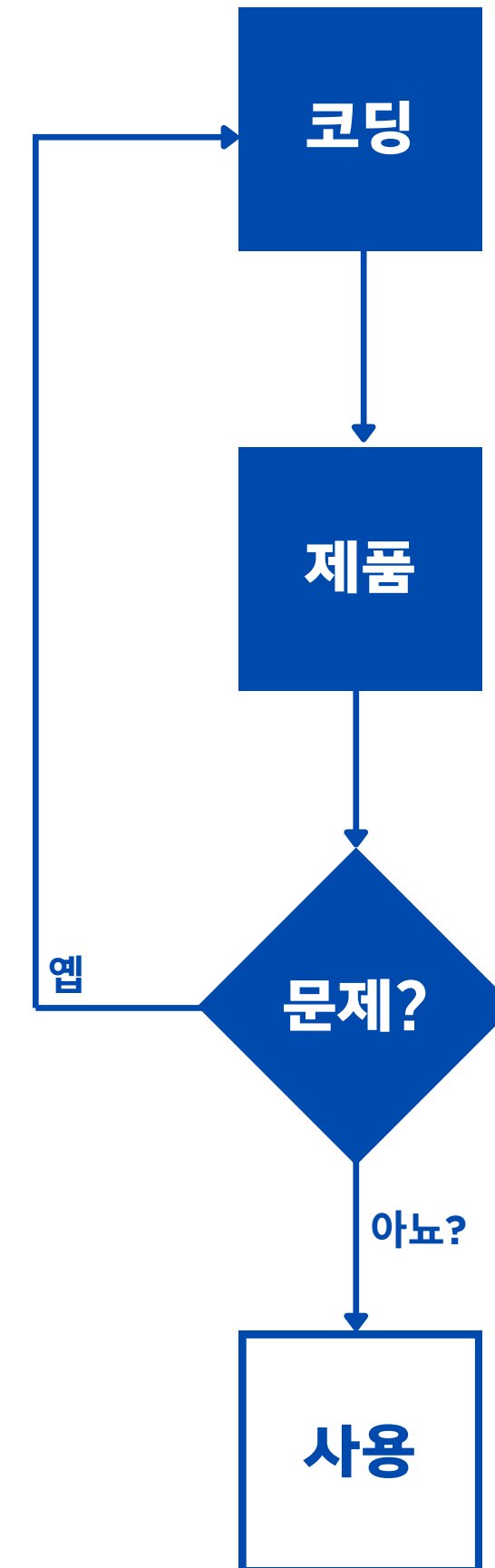
*EVOLUTIONARY PROCESS*

*SPIRAL*

*PHASED DEVELOPMENT*

*OBJECT MANAGEMENT GROUP*

*AGILE-SCRUM*

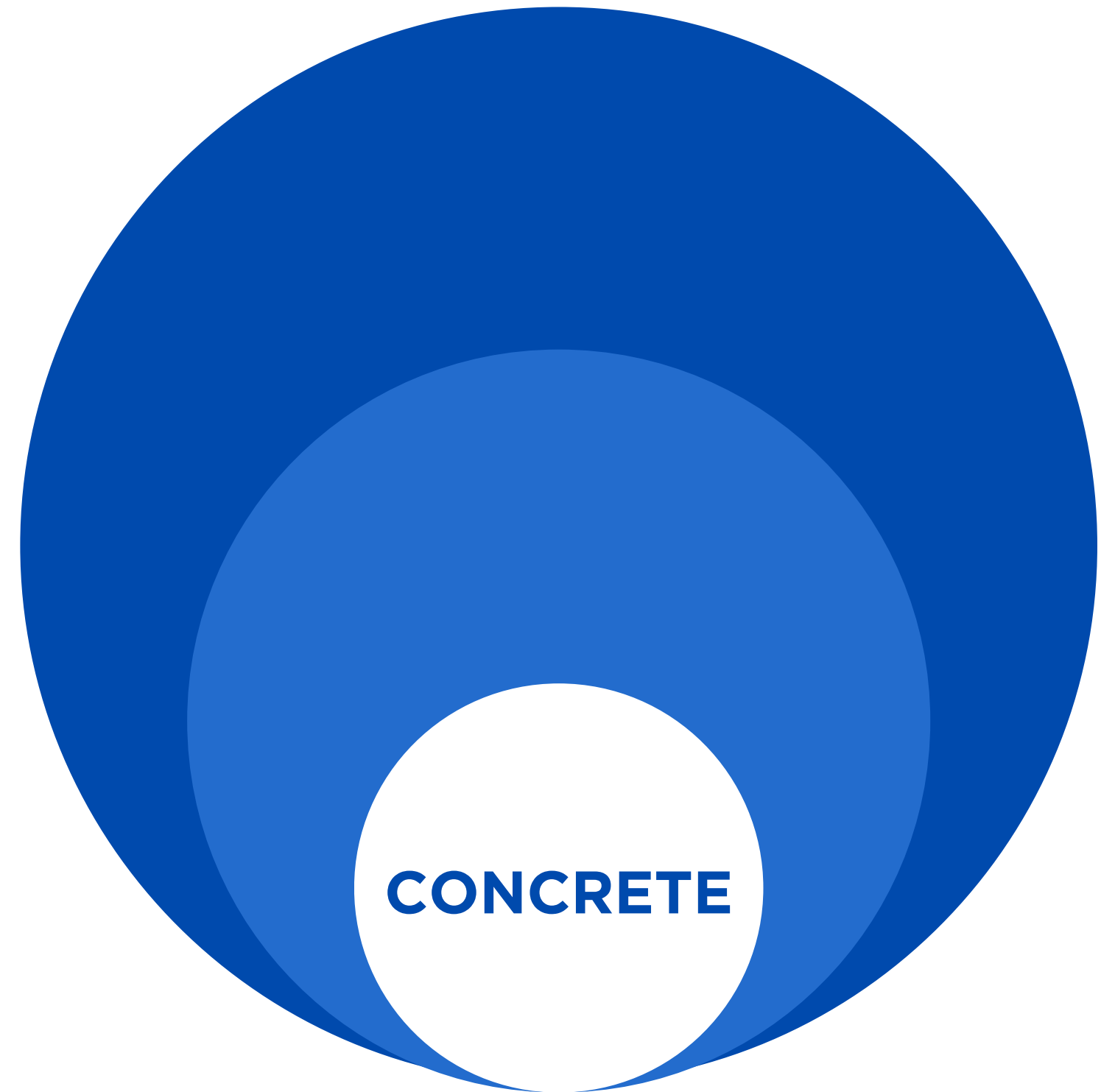


# 1. CONCRETE

## 실체적인 경험으로 시작하기

\*본 학습 방법은 CRA Approach를 기반으로 진행했습니다.  
자세한 내용은 사이트 참고

**개발 환경부터 차근차근 구축**  
**간단한 기능 하나를 끝까지 구현하기**  
**직접 만든 결과 공유 및 피드백**



## 2. REPRESENTATIONAL

개념적 이해와 도식화

*HOW TO BIND DATA IN VUE.JS*

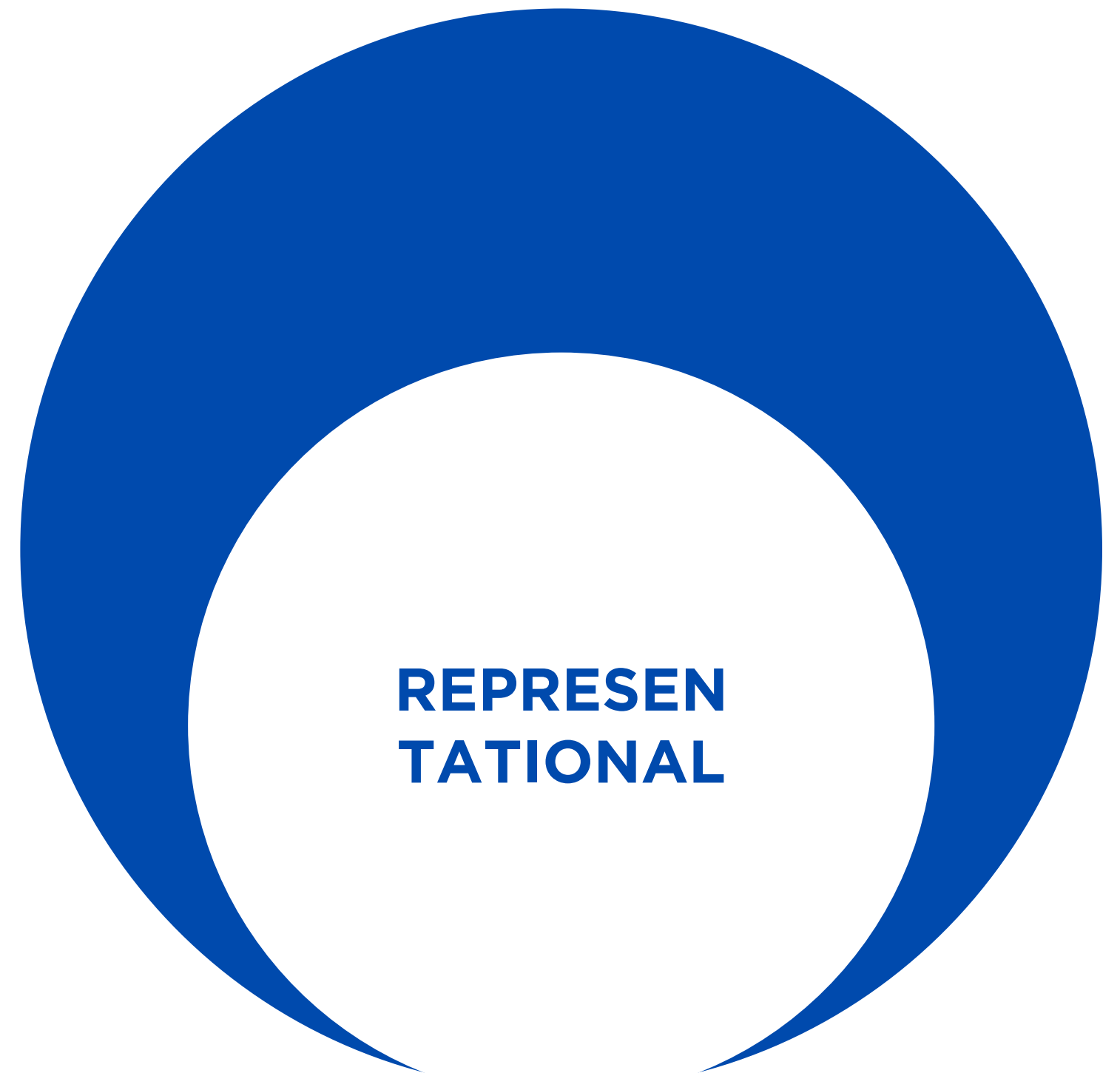
*WANNA KNOW REQUEST FLOW IN SPRING BOOT*

*LET'S FIND 'DOM' UPDATE*

**내부 동작 원리 분석**

**팀 내 지식 공유**

**문제 해결 과정 도식화**



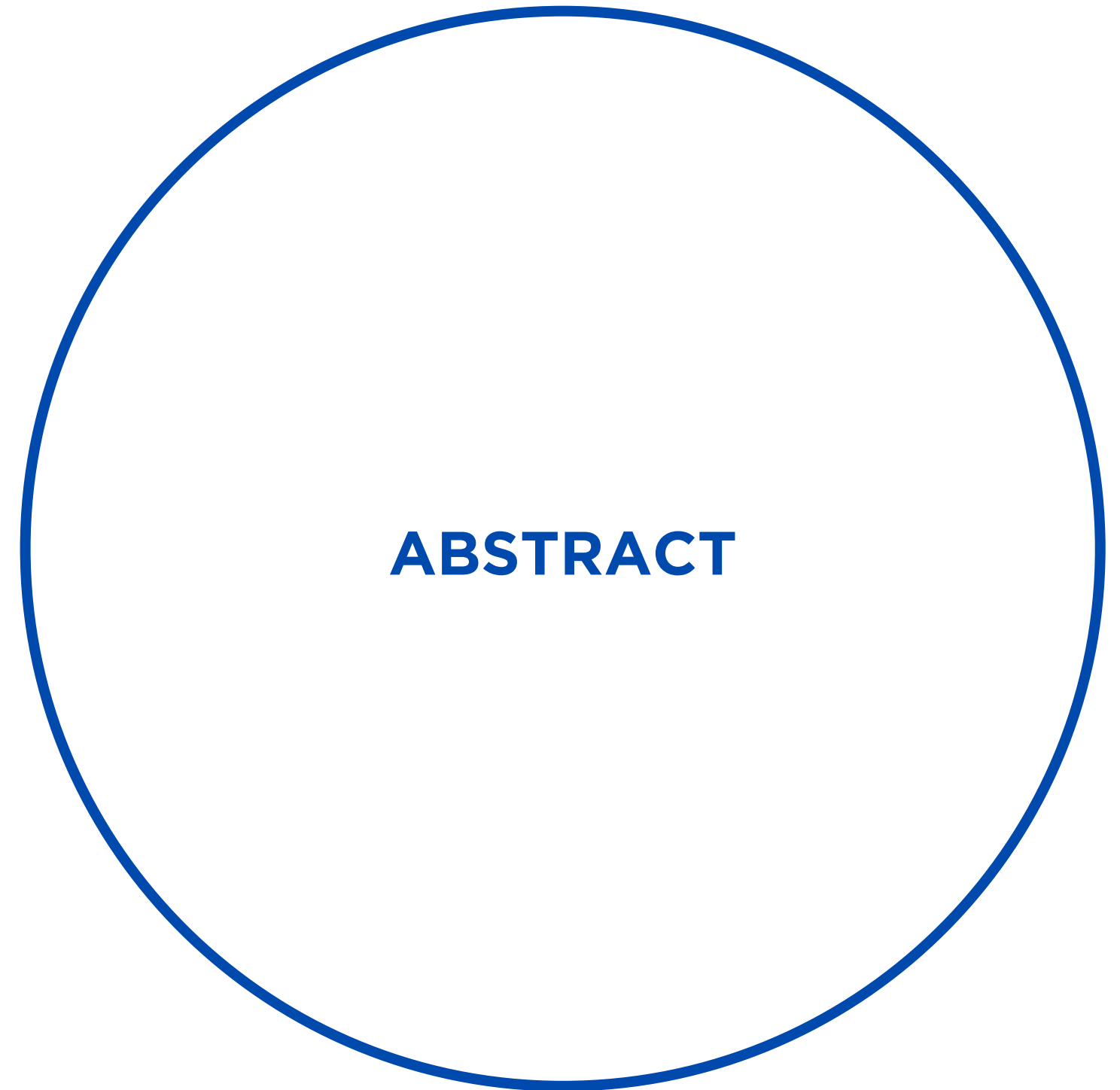
# 3. ABSTRACT

개념을 체계화하고 확장

*NOW IT'S TIME TO PROGRAM!*

**개념 체계화(직접 개념화/분야 개념 확립)**

**실제 프로젝트 확장**



**“해봐야 안다”**

# THANK YOU

**cas.aikorea@gmail.com**

**@CAS.KOREA**

**<https://caskorea.creatorlink.net/>**

