

# 프로젝트를 여러명이서 같이 개발할 때 생기는 문제점과 해결방법



안정민

# 소개

---

- 現 카카오뱅크 iOS 개발자 (16.10 ~ )
- 블로그 운영 : [minsone.github.io](https://minsone.github.io)

# 목차

---

- 프로젝트 거대화
- Xcode 프로젝트 관리
- CI/CD
- 아키텍처

# 프로젝트 거대화

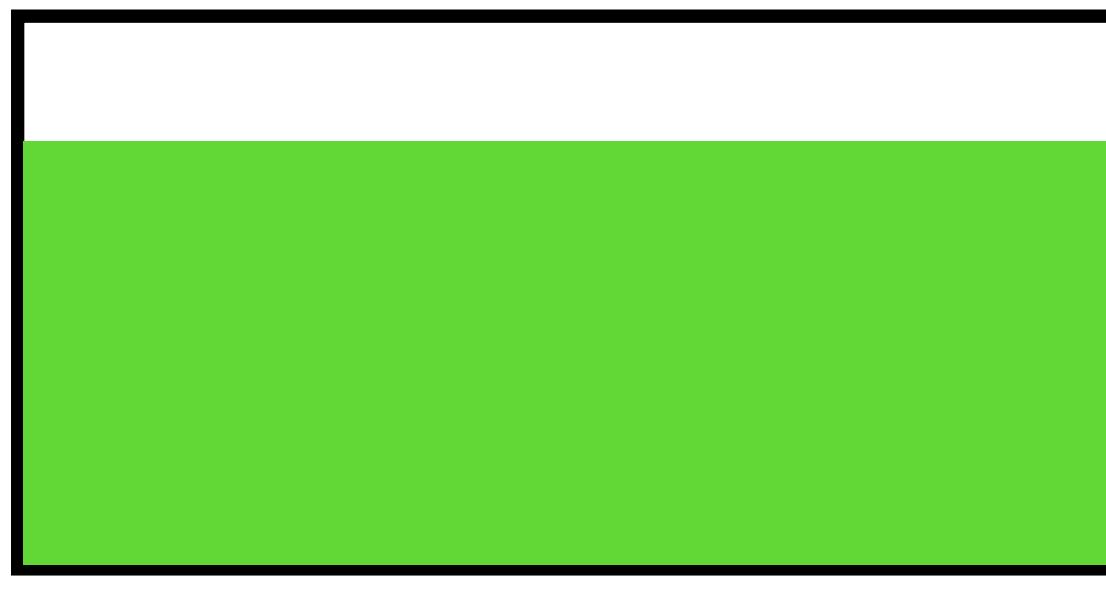
사업 규모

==

개발 인력 규모

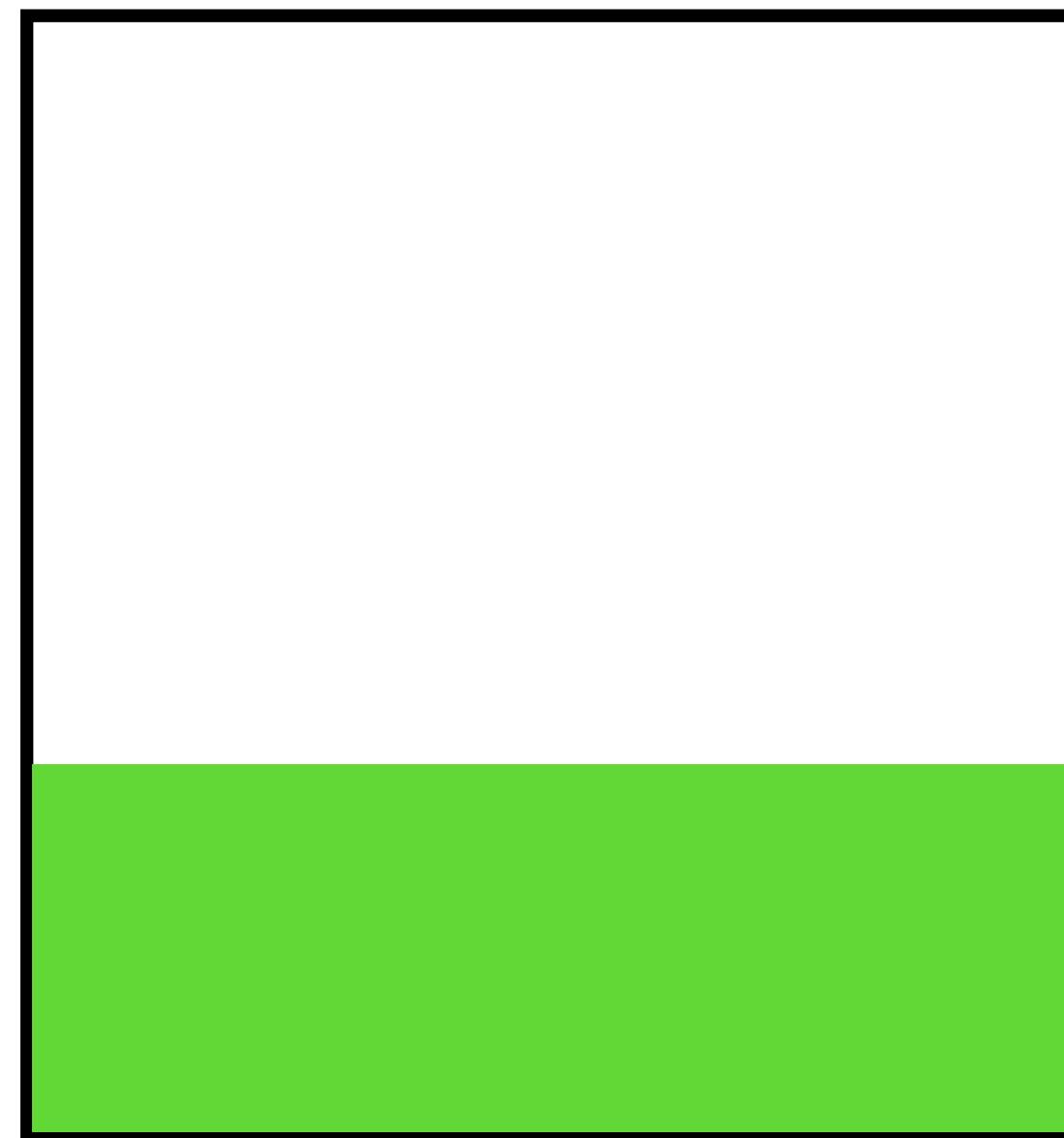
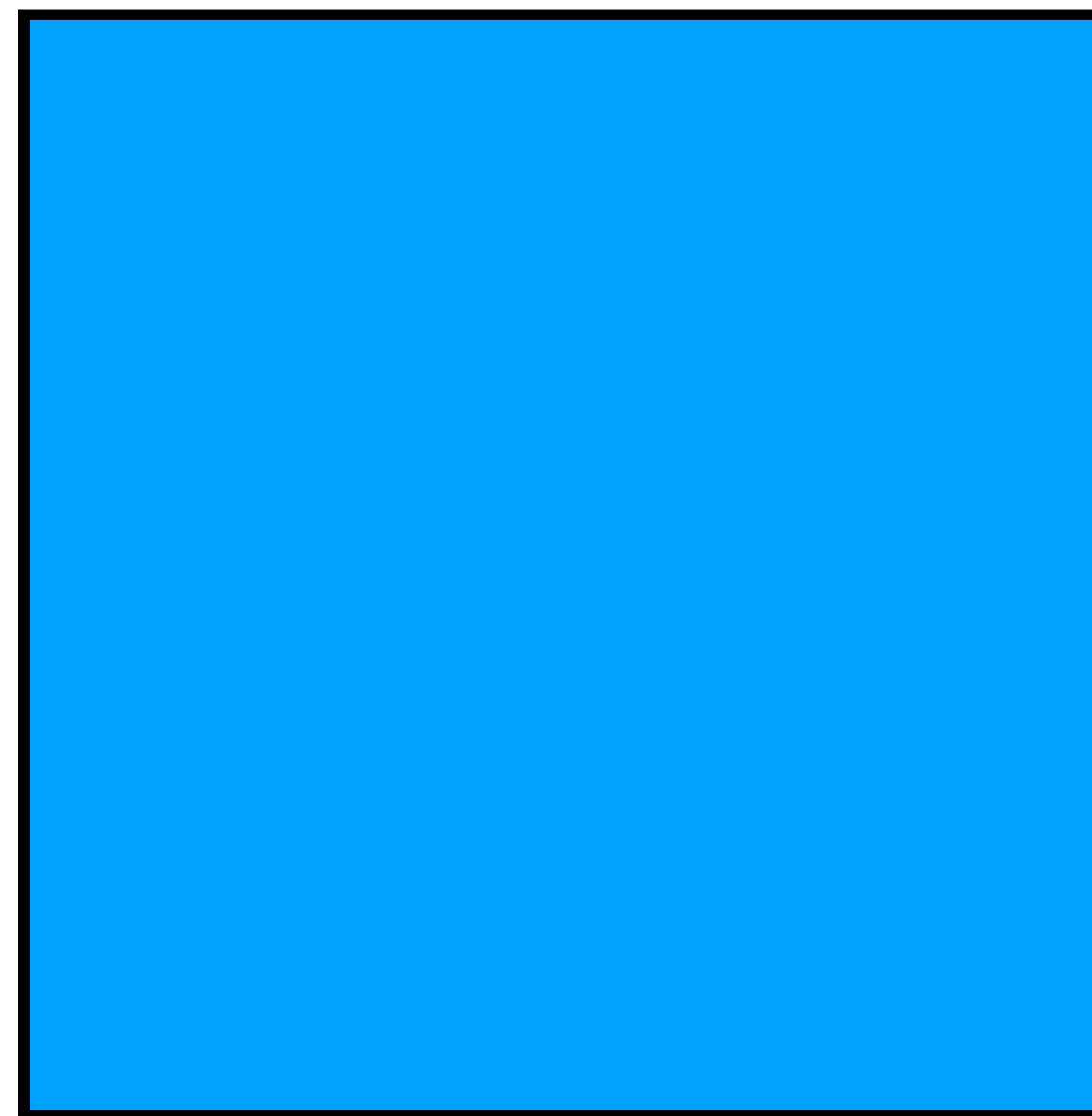
사업 규모

개발 인력 규모



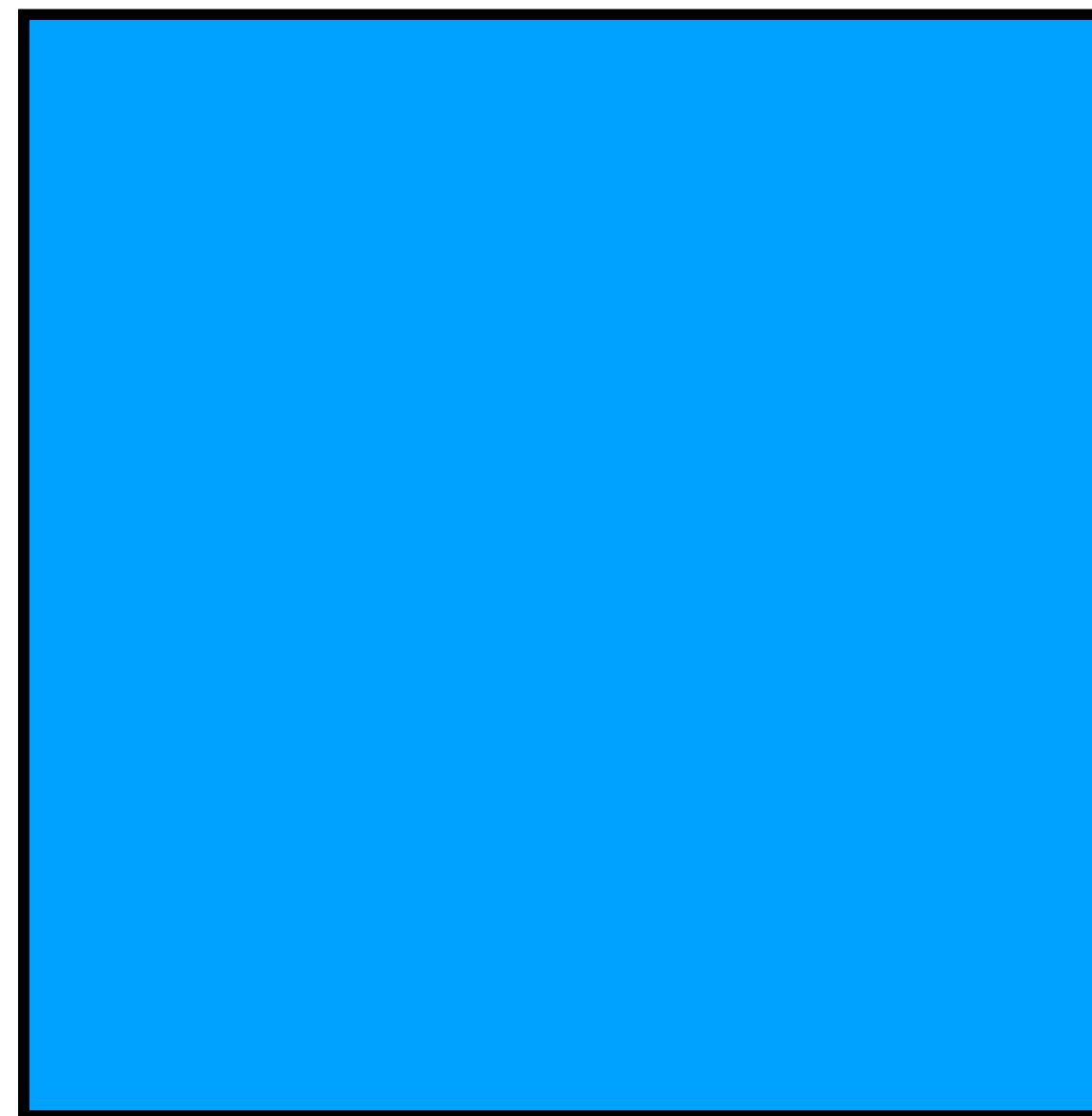
사업 규모

개발 인력 규모

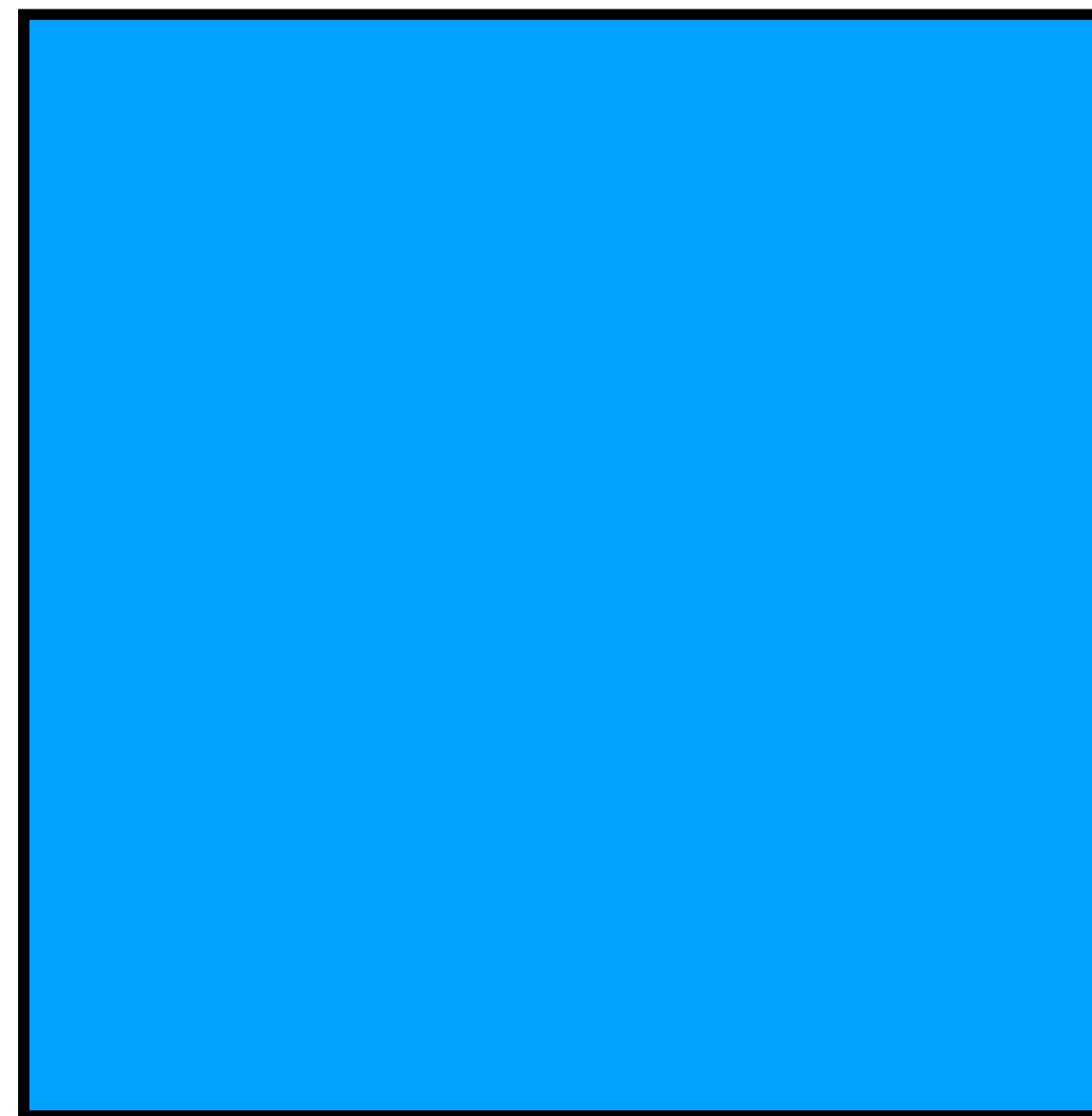


사업 규모

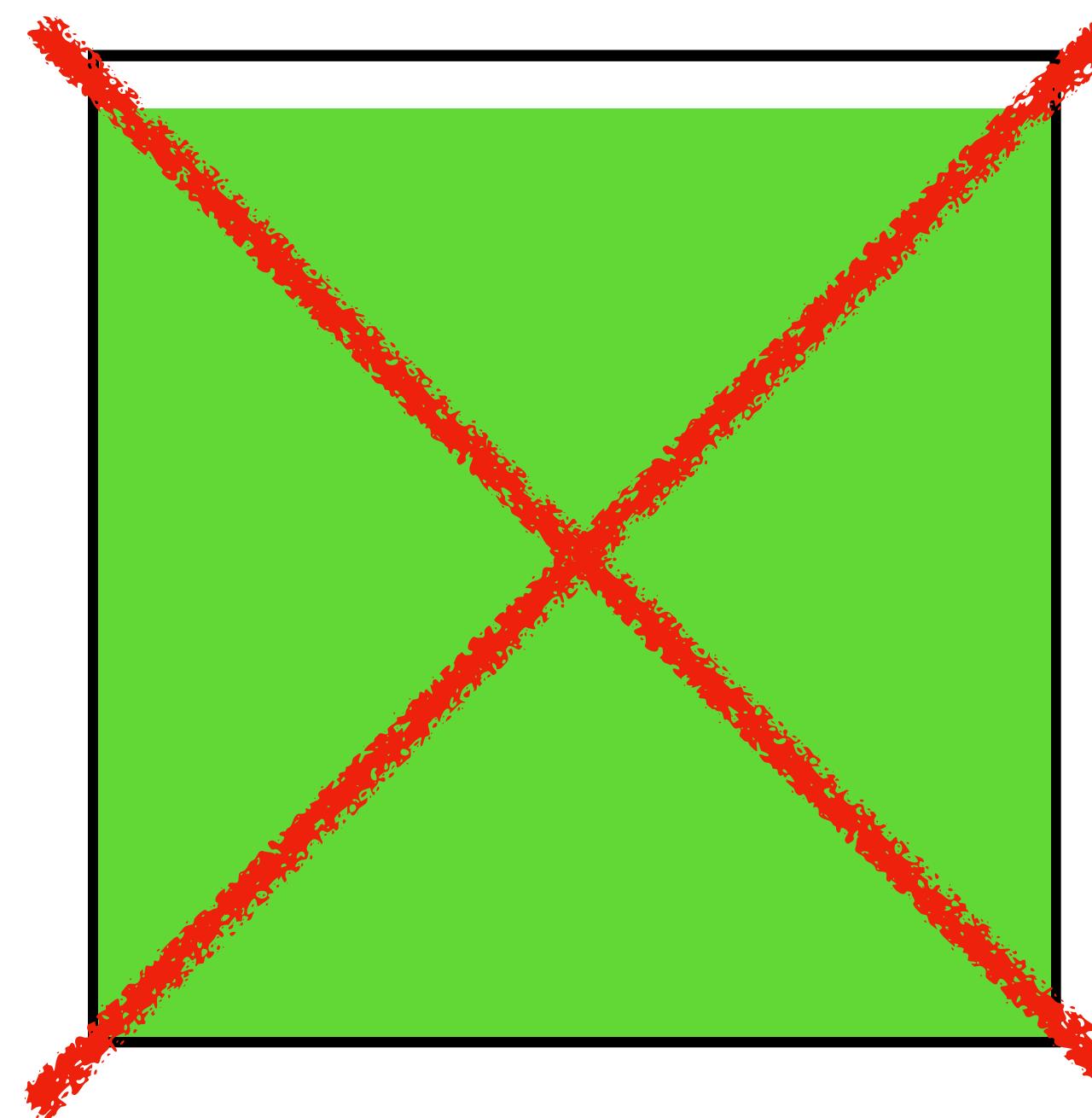
개발 인력 규모



사업 규모



개발 인력 규모



사업 규모

개발 인력 규모

생산성

인원

# Xcode 프로젝트 관리

# xcodeproj

- 프로젝트 설정을 관리하는 파일로 XML형식
- 타겟을 이용하여 각 환경에 맞춰 작업 가능



# **xcodeproj**

- 파일 생성시 프로젝트 및 인원이 많아질수록 빈번한 충돌 문제
- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.
- 모듈로 분리하지 않는 경우, 프로젝트 내에서 결합도가 빠른 속도로 증가

# xcodeproj

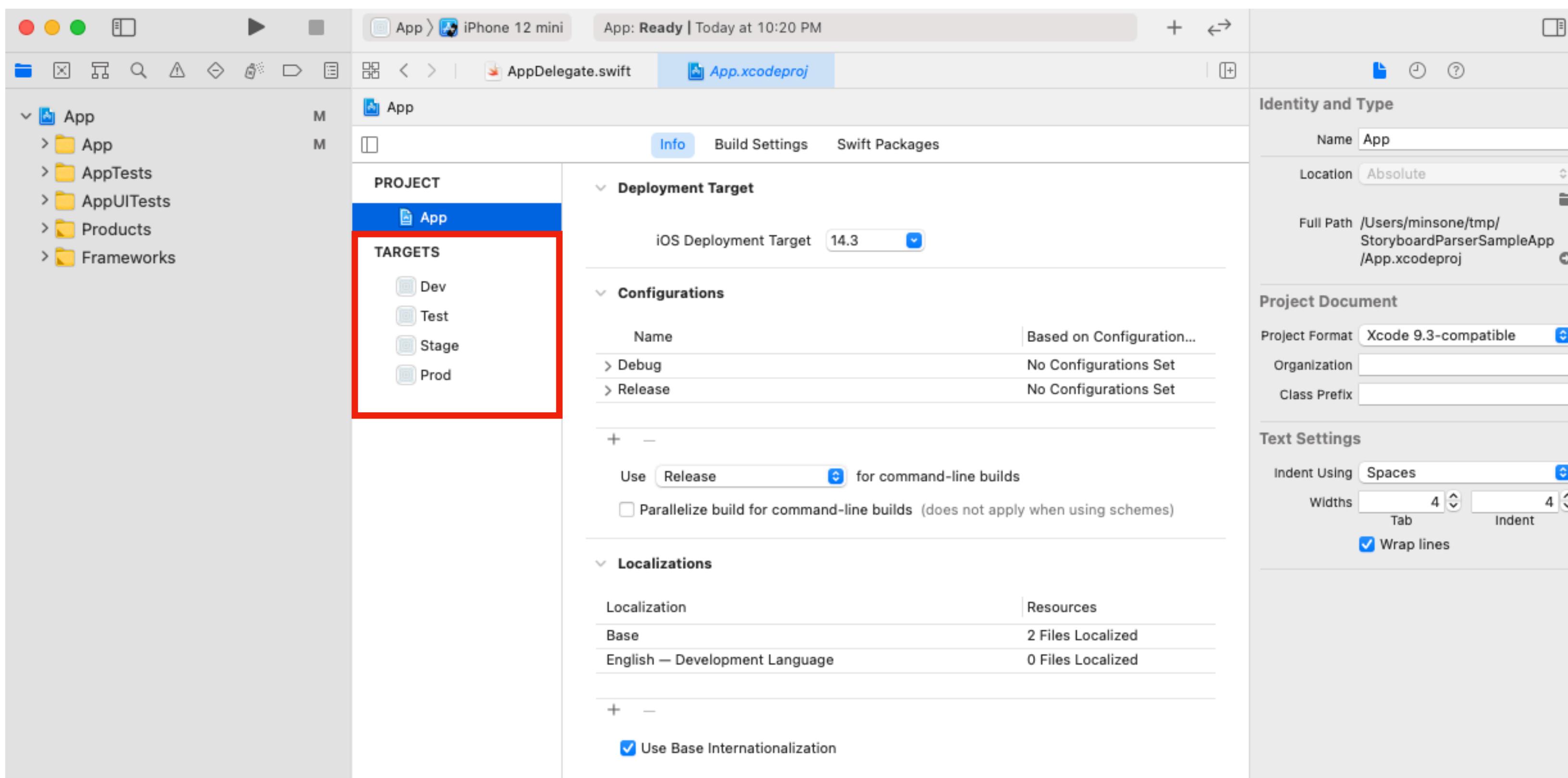
- 파일 생성시 프로젝트 및 인원이 많아질수록 빈번한 충돌 문제
- 인원이 많으면 코드, 파일 생산량 증가
  - > 프로젝트에 파일이 다수 추가됨
  - > 프로젝트 파일이 자주 수정되며, 파일 추가 또는 삭제로 인한 빈번한 충돌

# **xcodeproj**

- 파일 생성시 프로젝트 및 인원이 많아질수록 빈번한 충돌 문제
- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.
- 모듈로 분리하지 않는 경우, 프로젝트 내에서 결합도가 빠른 속도로 증가

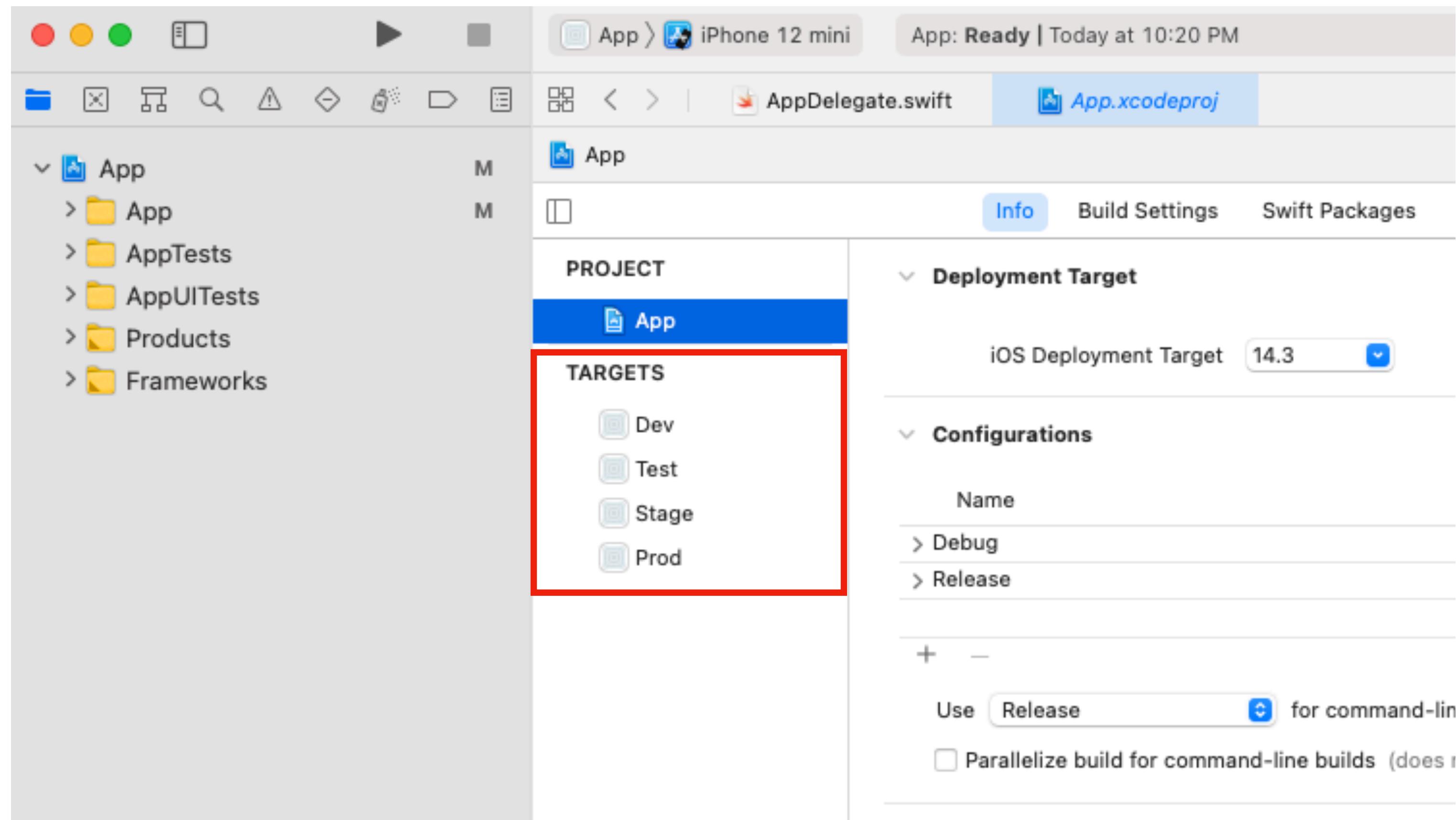
# xcodeproj

- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.



# xcodeproj

- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.



# xcodeproj

- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.
- 대부분의 타겟은 같은 파일을 가짐
  - > 타겟 1개 늘때마다 프로젝트 파일에 많은 내용이 추가됨
  - > 타겟이 늘어남에 따라 인덱싱 속도가 느려짐
  - > 파일 추가/삭제시 빈번한 충돌 발생

# xcodeproj

- 파일 생성시 프로젝트 및 인원이 많아질수록 빈번한 충돌 문제
- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.
- 모듈로 분리하지 않는 경우, 프로젝트 내에서 결합도가 빠른 속도로 증가

# xcodeproj

- 모듈로 분리하지 않는 경우, 프로젝트 내에서 결합도가 빠른 속도로 증가
- 프로젝트 내에선 public, internal이 같은 개념.
  - > 도메인이 다른 코드 등을 어디서든 가져다 사용하거나 정의할 수 있음(격리화 불가)
  - > 각 도메인과의 결합이 생길 가능성 증가
  - > 다수가 작업시 결합도가 빠른 속도로 증가
  - > 각 기능을 모듈로 분리할 때 코드 분리가 어려움.

# xcodeproj

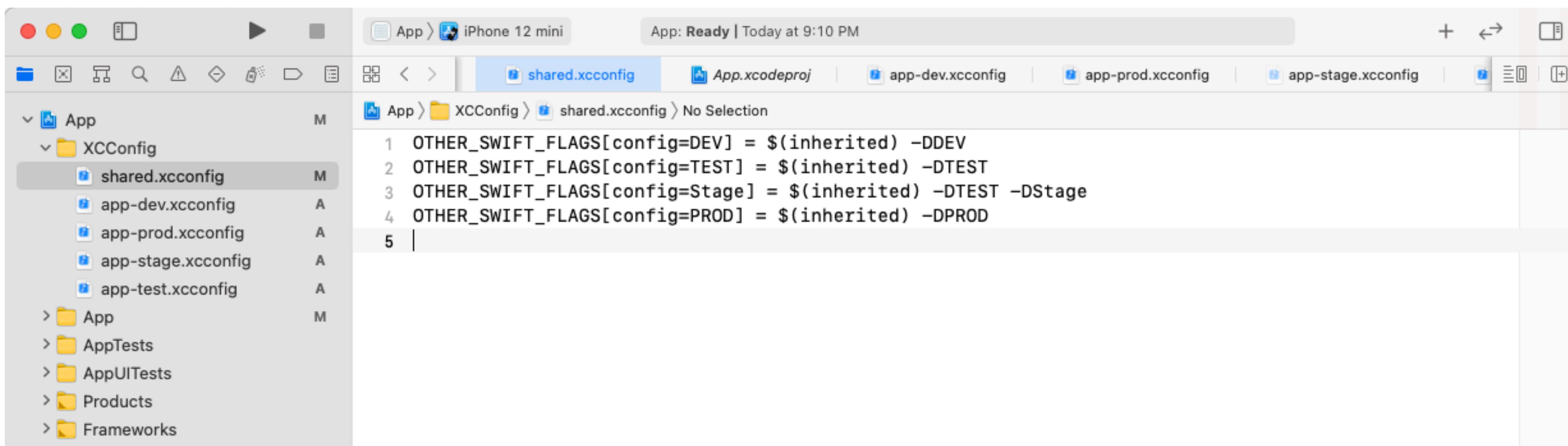
- 파일 생성시 프로젝트 및 인원이 많아질수록 빈번한 충돌 문제
- 각 개발 환경을 타겟으로 관리하는 경우 프로젝트가 커질수록 프로젝트 인덱싱이 느려짐.
- 모듈로 분리하지 않는 경우, 프로젝트 내에서 결합도가 빠른 속도로 증가

# **xcodeproj**

- XCConfig를 적용하여 프로젝트 설정을 관리
- 적용한 XCConfig를 토대로 타겟을 정리(1~2개)
- 각 기능마다 별도의 프로젝트로 분리로 코드 응집도 증가 및 결합도 감소

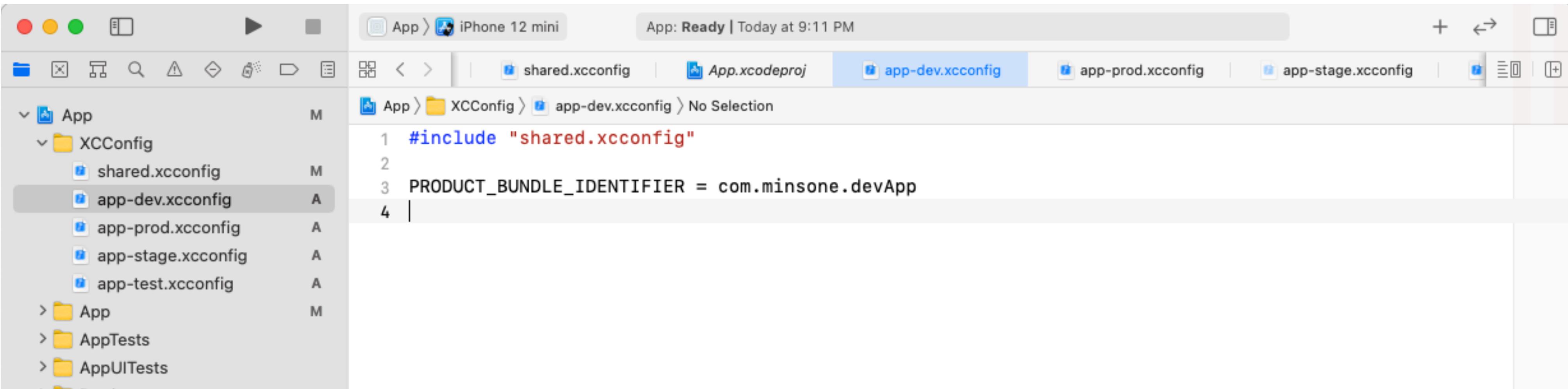
# xcodeproj

- XCConfig를 적용하여 프로젝트 설정을 관리
- 공통 XCConfig를 생성



# xcodeproj

- XCConfig를 적용하여 프로젝트 설정을 관리
- 공통 XCConfig를 생성
  - > 각 타겟에 맞는 XCConfig를 생성
  - > 각 XCConfig는 공통 XCConfig를 include하여 공통 설정을 따르도록 함.
  - > 각 XCConfig는 변경할 환경 설정, 추가할 설정 값을 할당.



The screenshot shows the Xcode interface with the project navigation bar at the top. The selected tab is 'app-dev.xcconfig'. The left sidebar shows the project structure with an 'App' folder containing an 'XCConfig' folder. Inside 'XCConfig', there are five files: 'shared.xcconfig', 'app-dev.xcconfig' (which is selected), 'app-prod.xcconfig', 'app-stage.xcconfig', and 'app-test.xcconfig'. The main editor area displays the contents of 'app-dev.xcconfig':

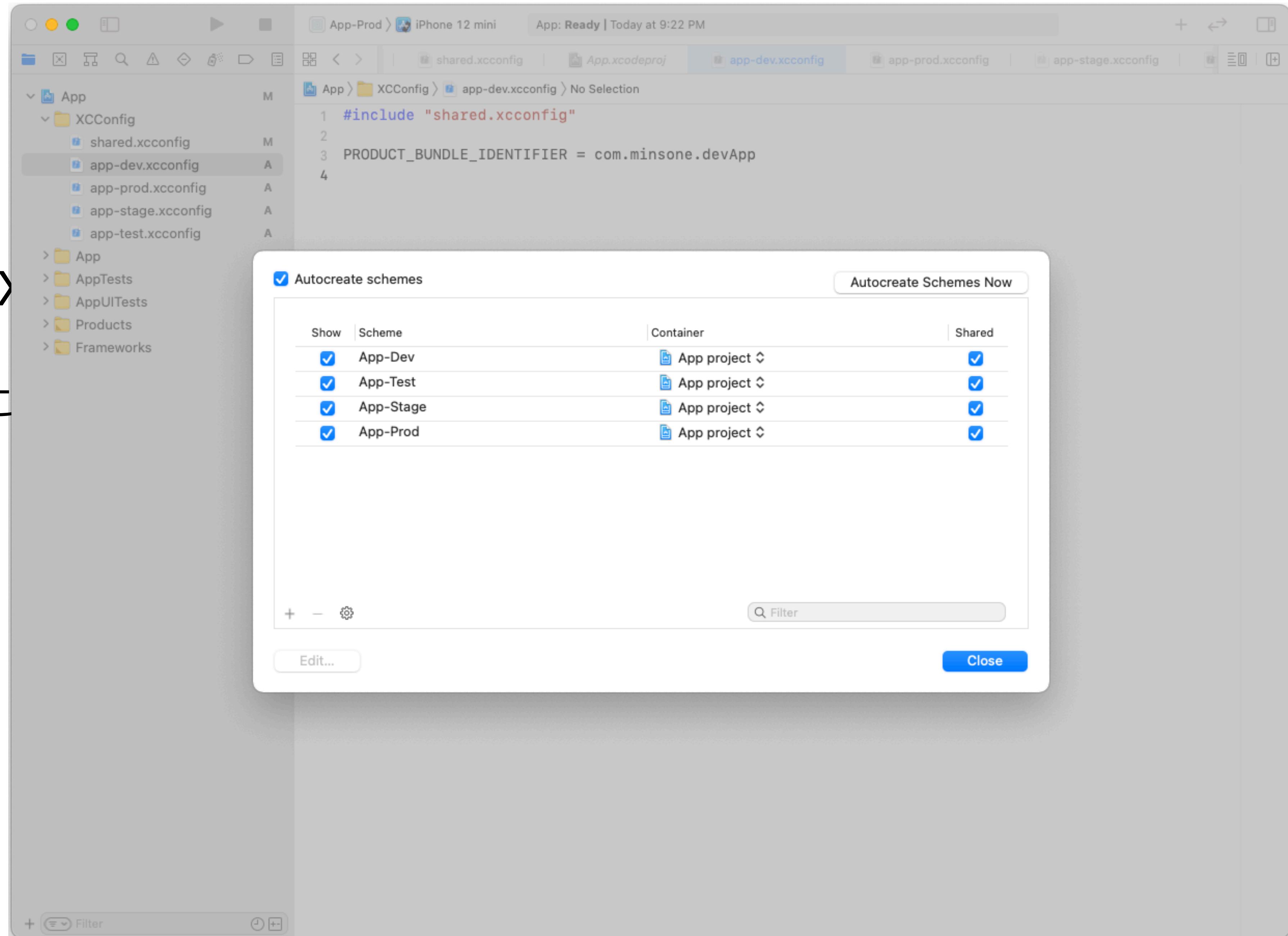
```
1 #include "shared.xcconfig"
2
3 PRODUCT_BUNDLE_IDENTIFIER = com.minsone.devApp
4
```

# **xcodeproj**

- XCConfig를 적용하여 프로젝트 설정을 관리
- 적용한 XCConfig를 토대로 타겟을 정리(1~2개)
- 각 기능마다 별도의 프로젝트로 분리로 코드 응집도 증가 및 결합도 감소

# **xcodeproj**

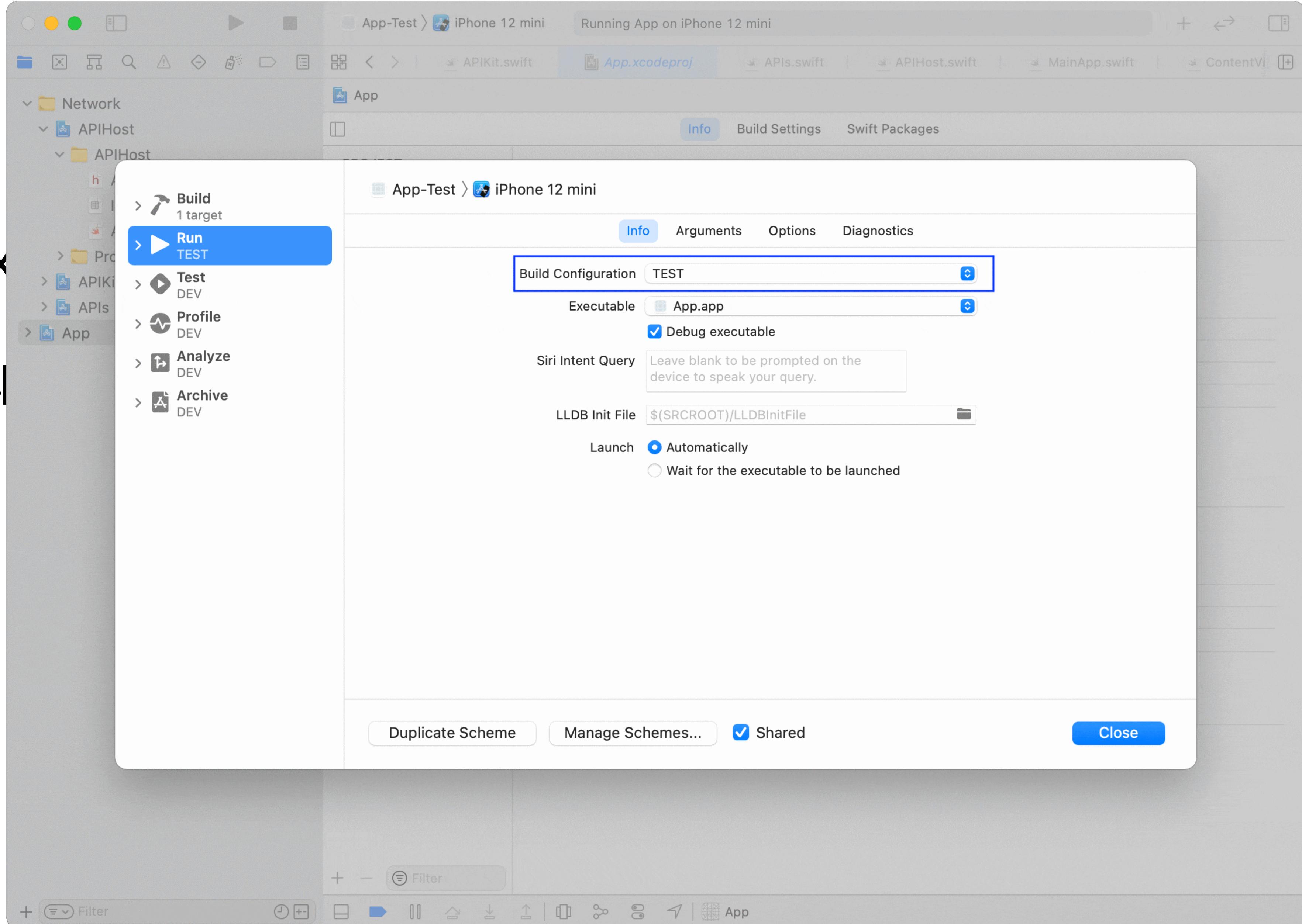
- 적용한 XCConfig를 토대로 타겟을 정리(1~2개)
  - 타겟 대신 스키마(Scheme)을으로 대신하고 Build Configuration을 맞춤.



● 적용한 >

● 타겟 디스플레이

- 적용한 X
- 타겟 디

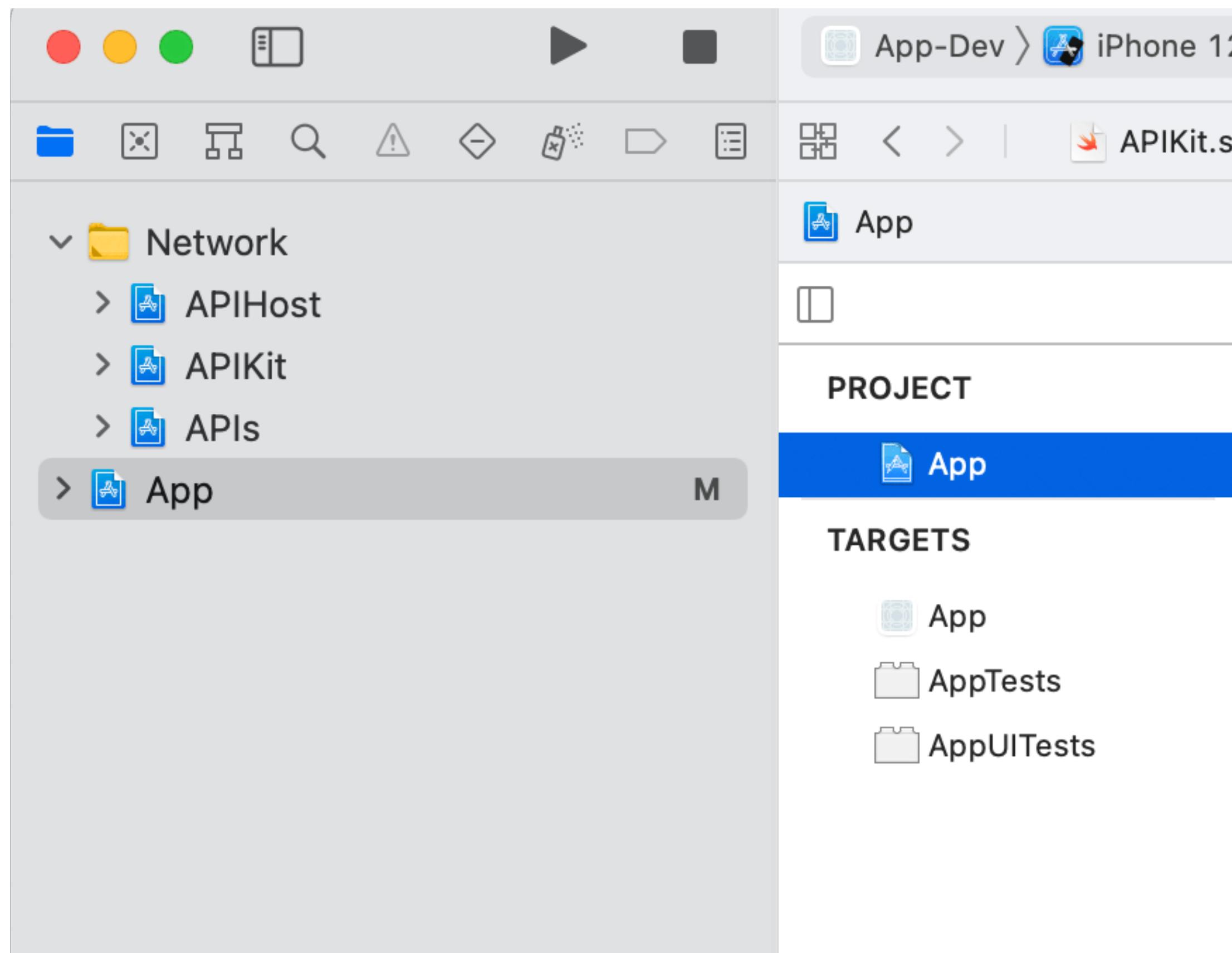


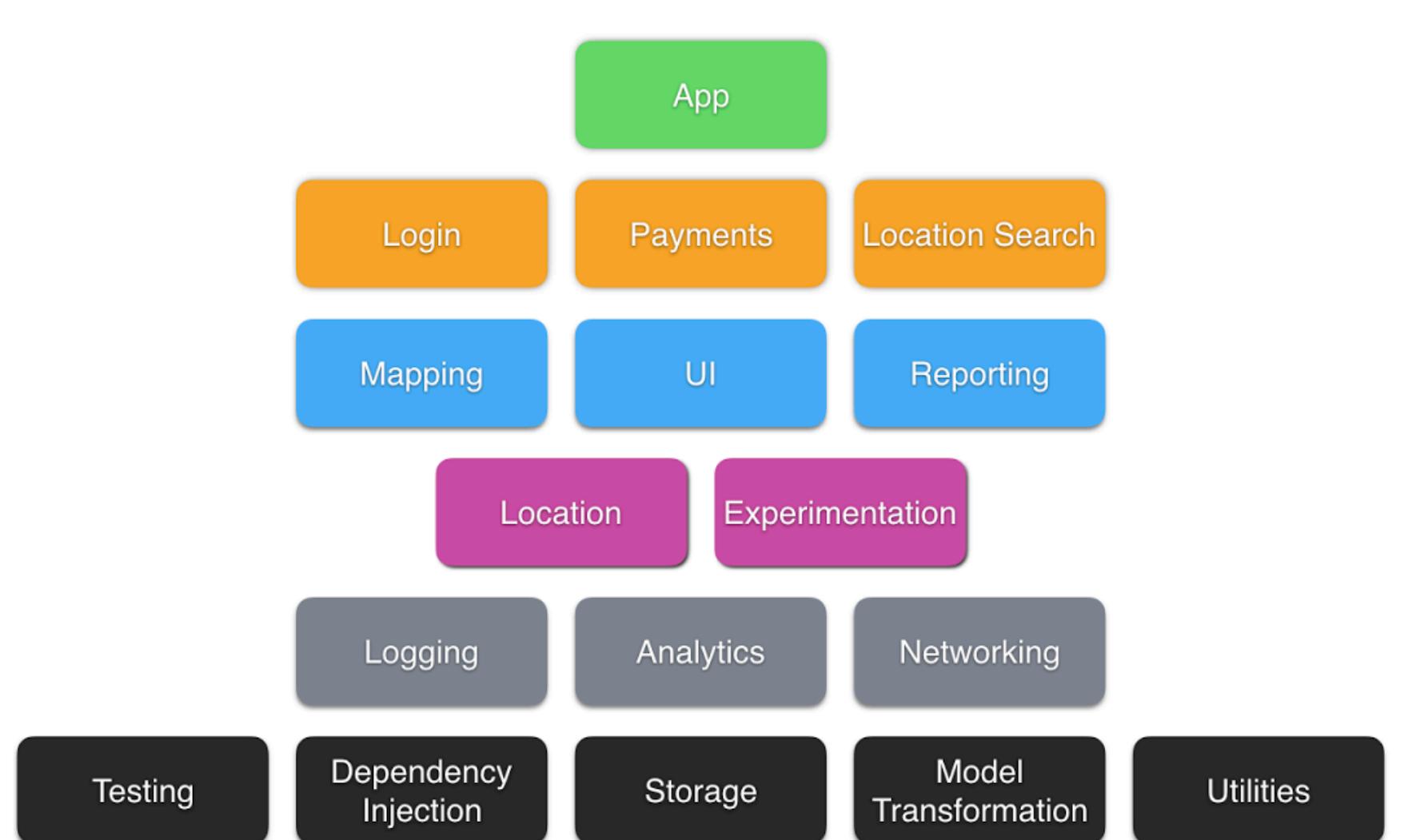
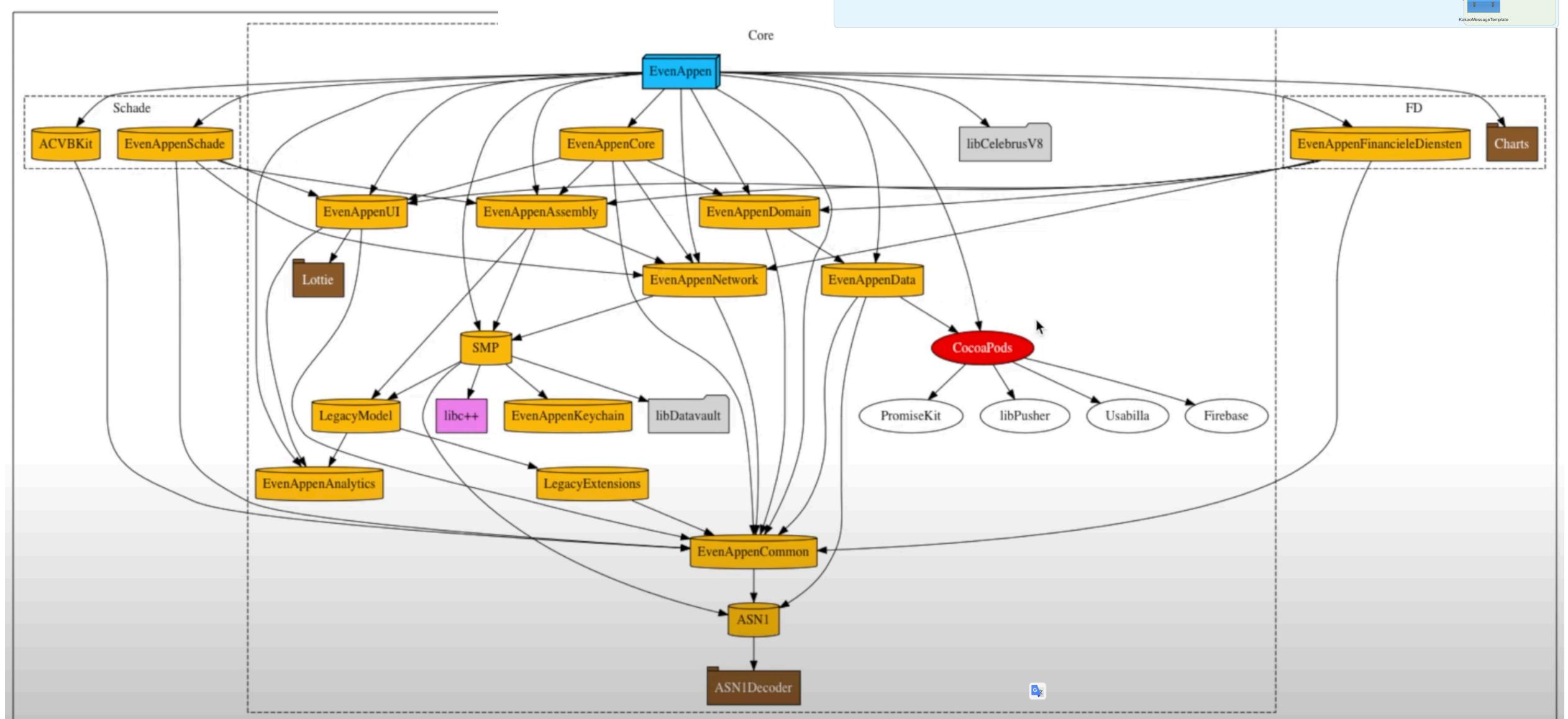
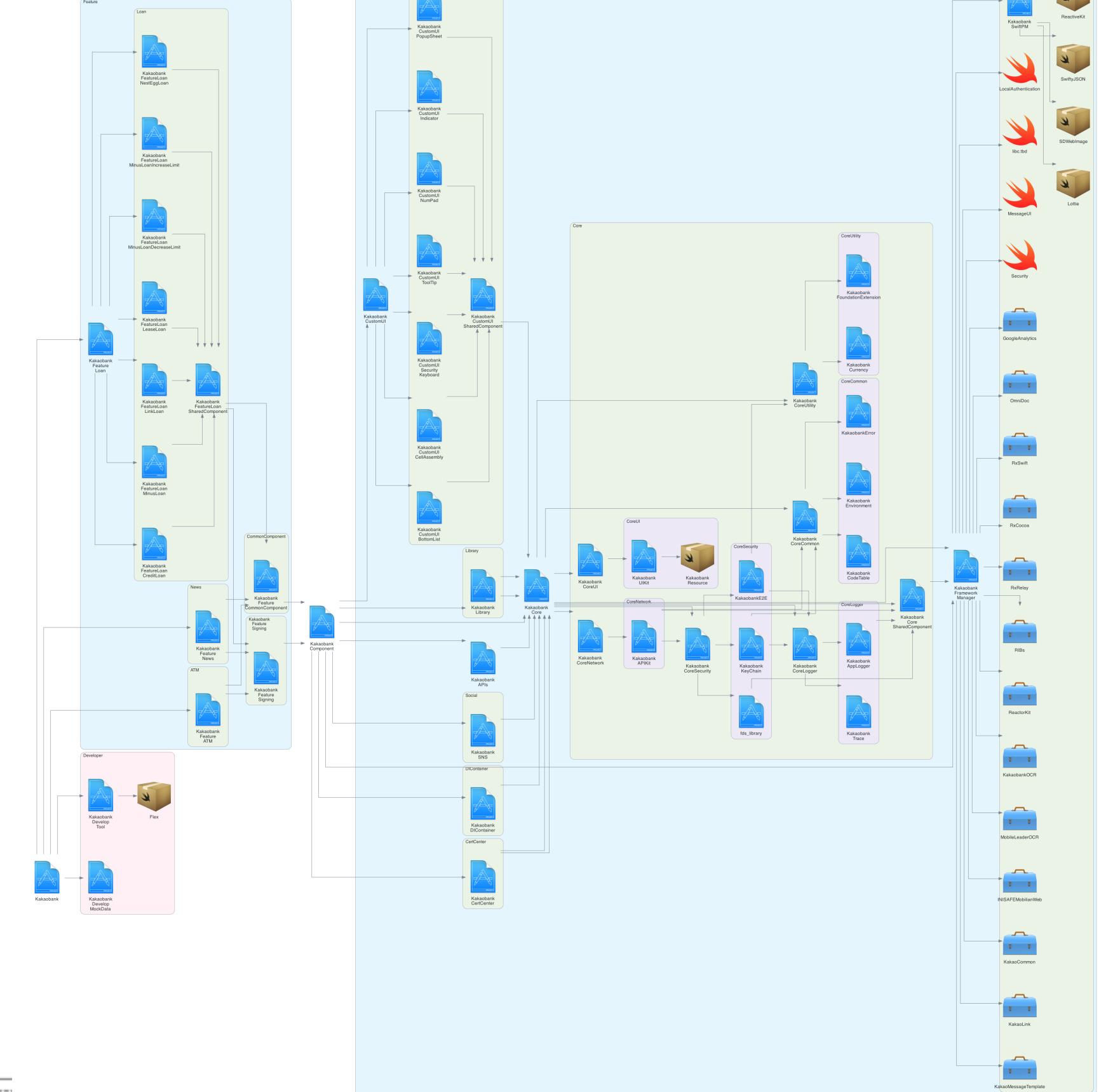
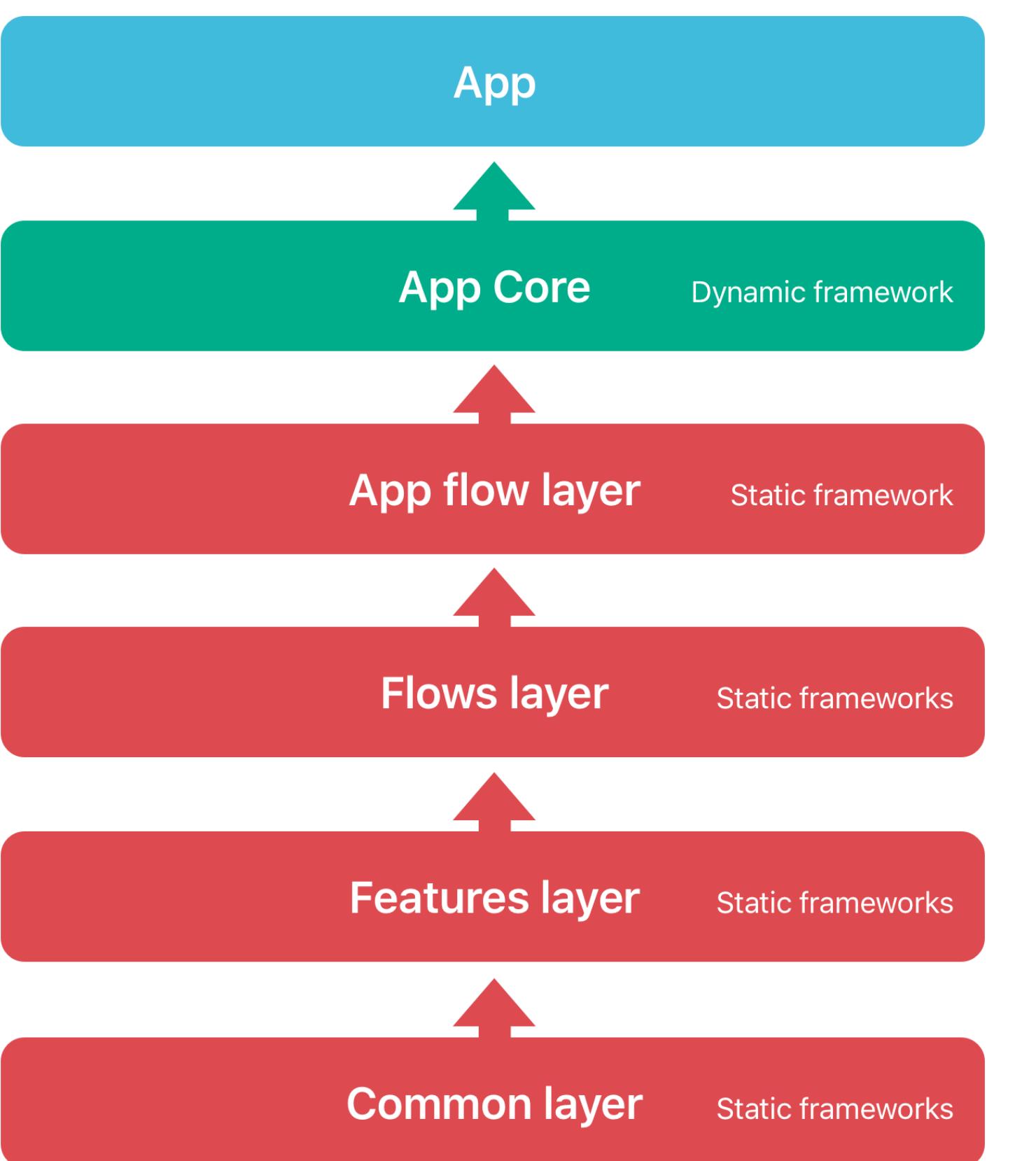
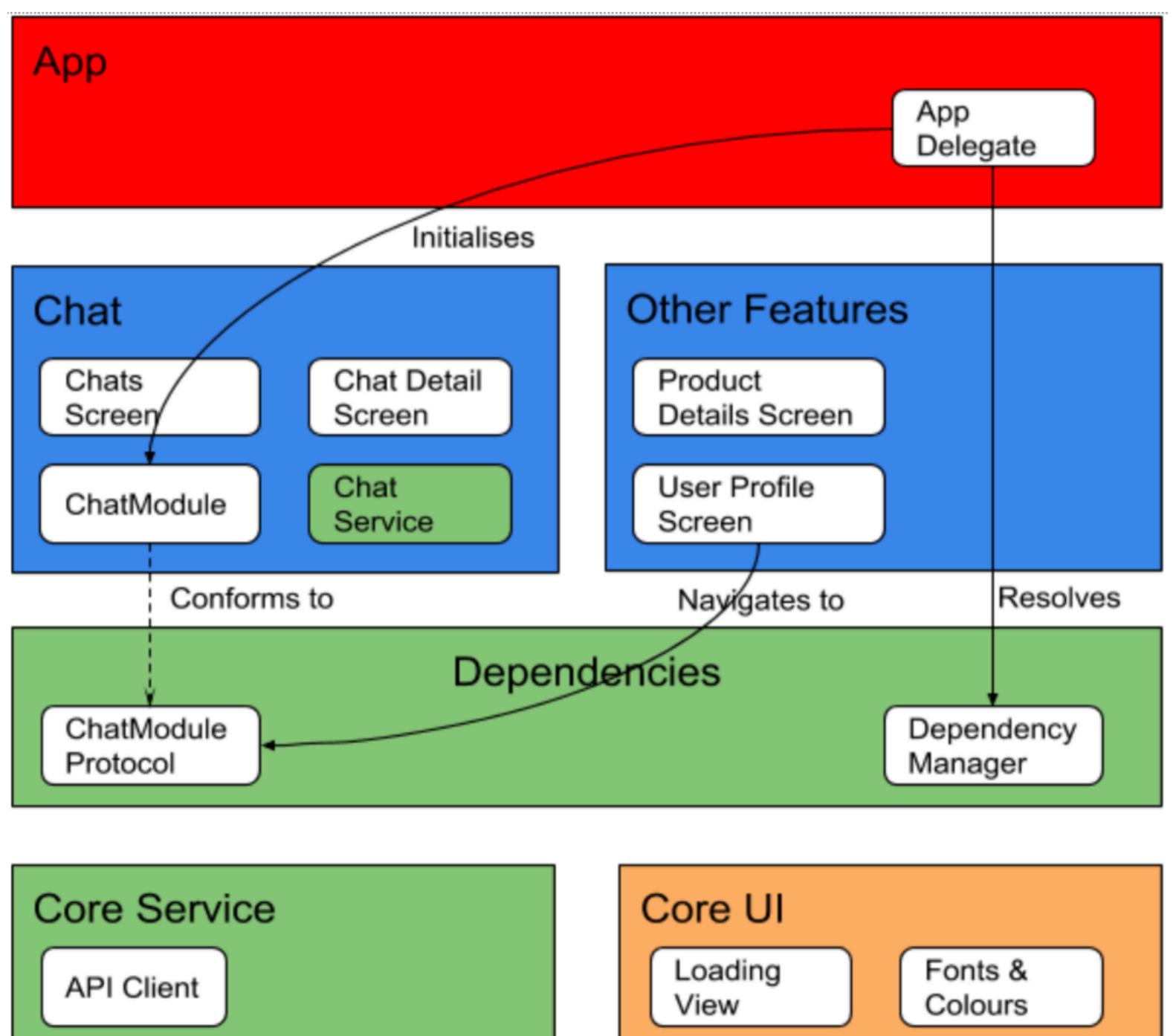
# **xcodeproj**

- XCConfig를 적용하여 프로젝트 설정을 관리
- 적용한 XCConfig를 토대로 타겟을 정리(1~2개)
- 각 기능마다 별도의 프로젝트로 분리로 코드 응집도 증가 및 결합도 감소

# xcodeproj

- 각 기능마다 별도의 프로젝트로 분리로 코드 응집도 증가 및 결합도 감소





- Dynamic Library, Static Library 이해
- Swift Package 이해
- Linker 이해
- 모듈간의 Dependency 이해(Circular dependency)
- 바이너리 분석 방법 이해
- 어플리케이션 구동 절차 이해

# Continuous Integration Continuous Delivery

# Continuous Integration

- 통합된 코드에서 자신의 코드가 제대로 동작(빌드/테스트)가 되는지 확인
  - 언제든지 현재 코드가 배포나가더라도 문제가 없어야 함.
- 코드리뷰 필수
  - 자신의 코드가 머지되면 모두의 코드가 되기 때문에 크로스체크가 반드시 필요.

# Continuous Delivery

- 어플리케이션이 항상 신뢰 가능한 수준에서 배포될 수 있도록 관리.
  - 빌드 및 테스트가 지속적으로 일어나므로 배포 또한 지속적으로 이루어져야 함.
  - 배포의 어려움이 줄어들고, 재작업 시간이 점점 줄어듬.
- 프로젝트가 커질수록 기능별 배포가 요구됨.
- CI/CD를 전담할 수 있는 인력이 필요함.

아키텍처

# 아키텍처

- 템플릿 제공하는 아키텍처
- 기능을 세분화해서 개발 후 유연한 결합할 수 있는 아키텍처
- 테스트코드 작성이 가능한 아키텍처

끌  
끌