Xcode프로젝트 Tuist 도입하기

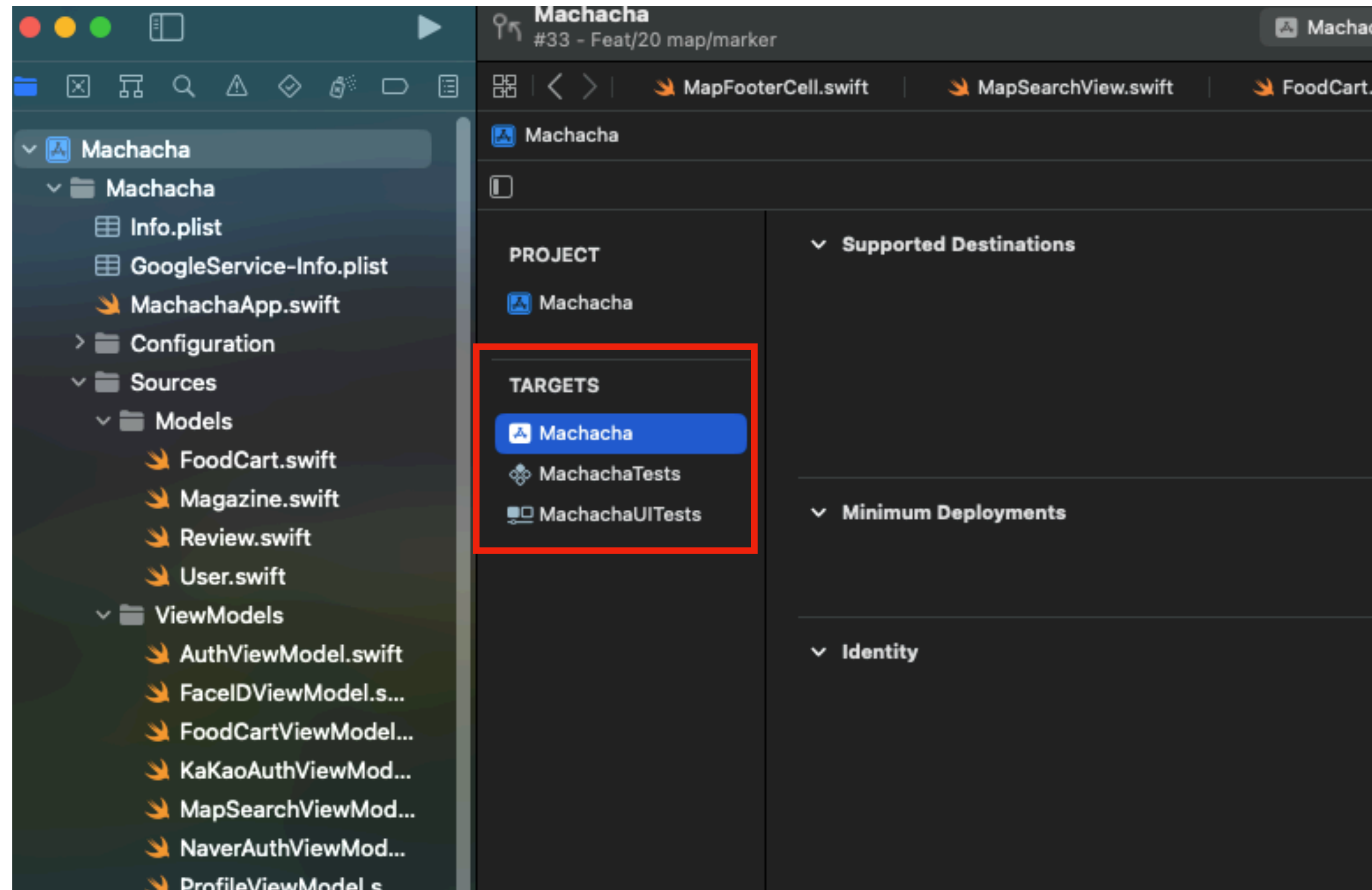박정우 2021.02.02

Tuist란?

Xcode 프로젝트의 생성, 유지 관리 및 상호 작용을 도와주는 CLI 도구
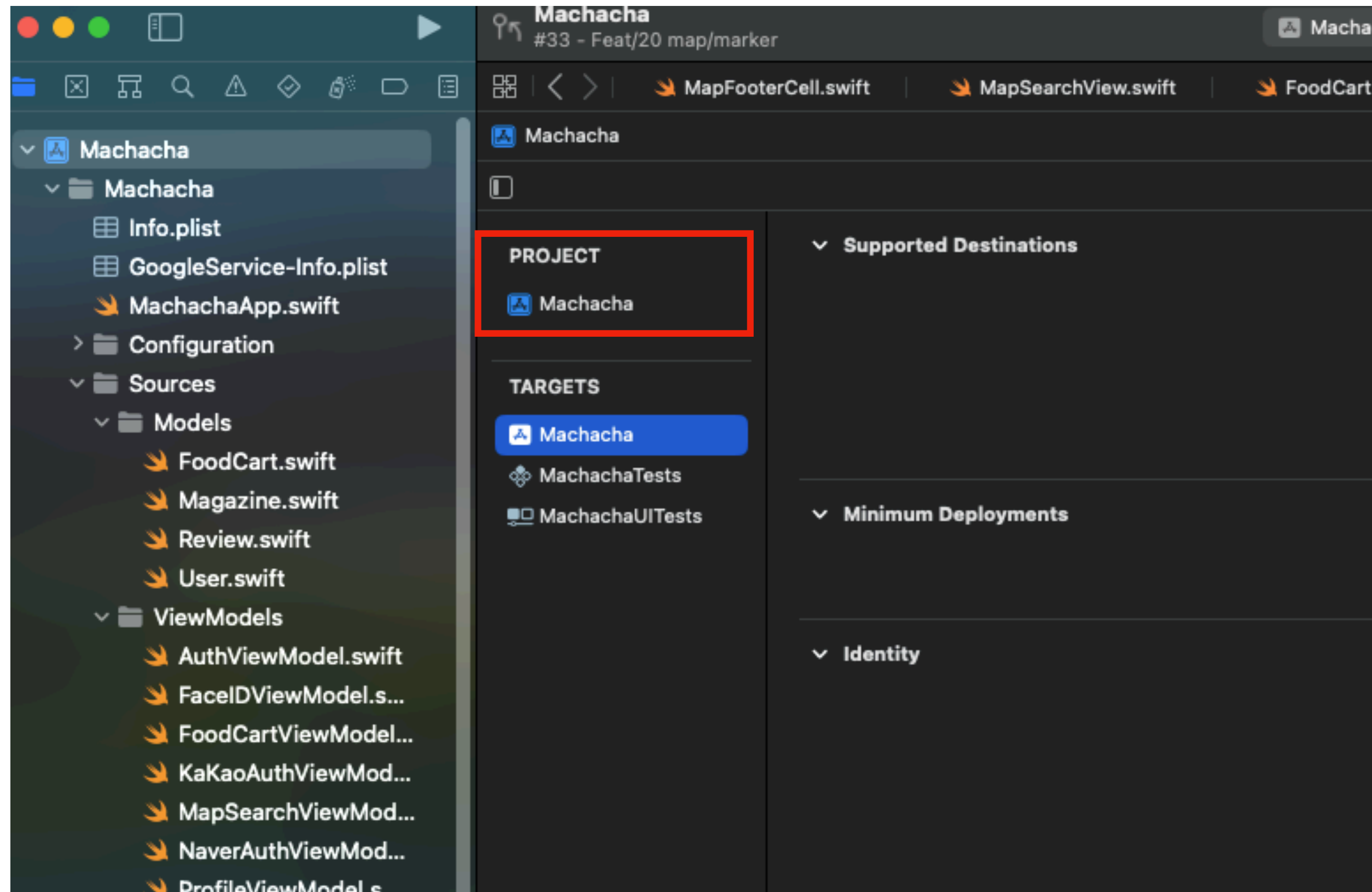
Tuist를 사용하기 전에..

- Target

- Project

- Workspace

- 모듈화

Target

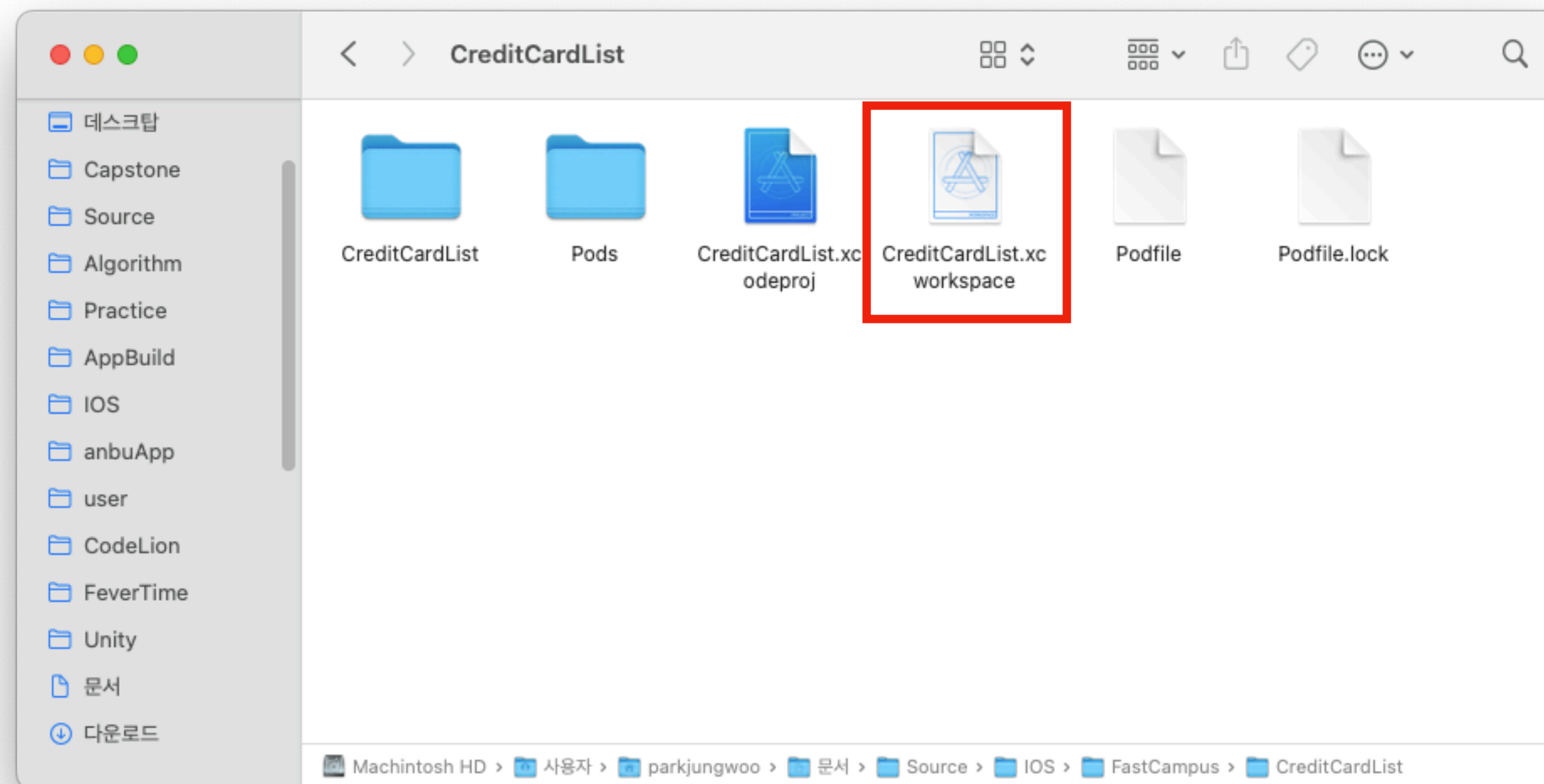project나 Workspace의 파일들을 빌드하여 생성되는 End Product를 의미
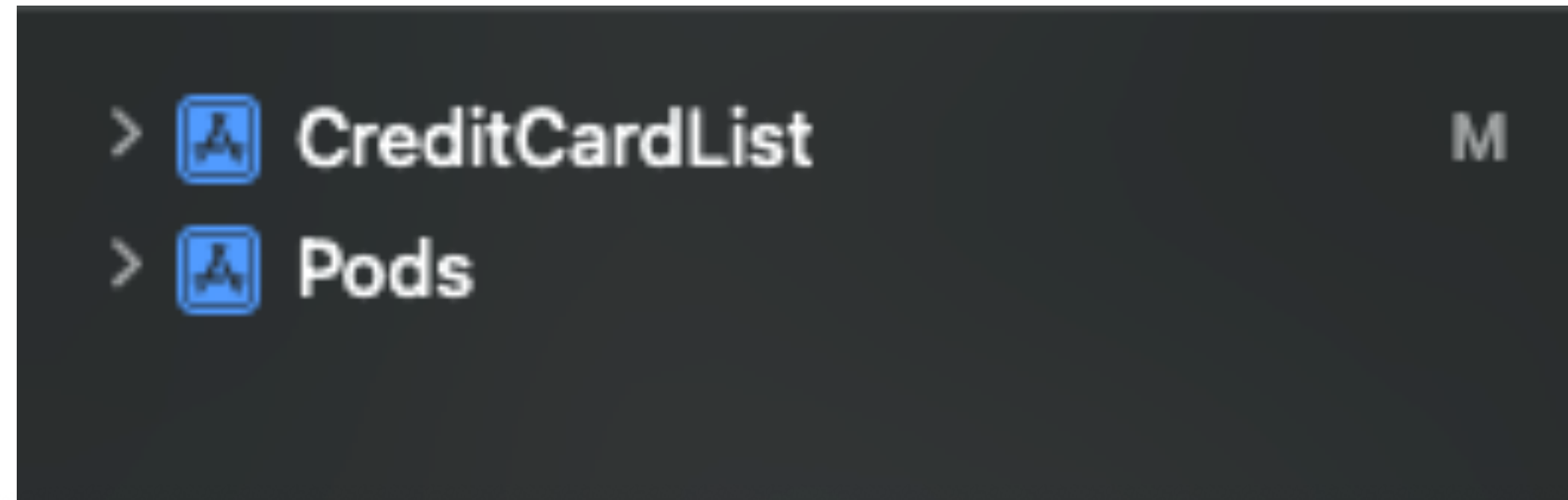
Project

Project가 가지고 있는 정보

- 소스파일에 대한 참조

- Debug, release와 같은 build configuration을 설정 가능
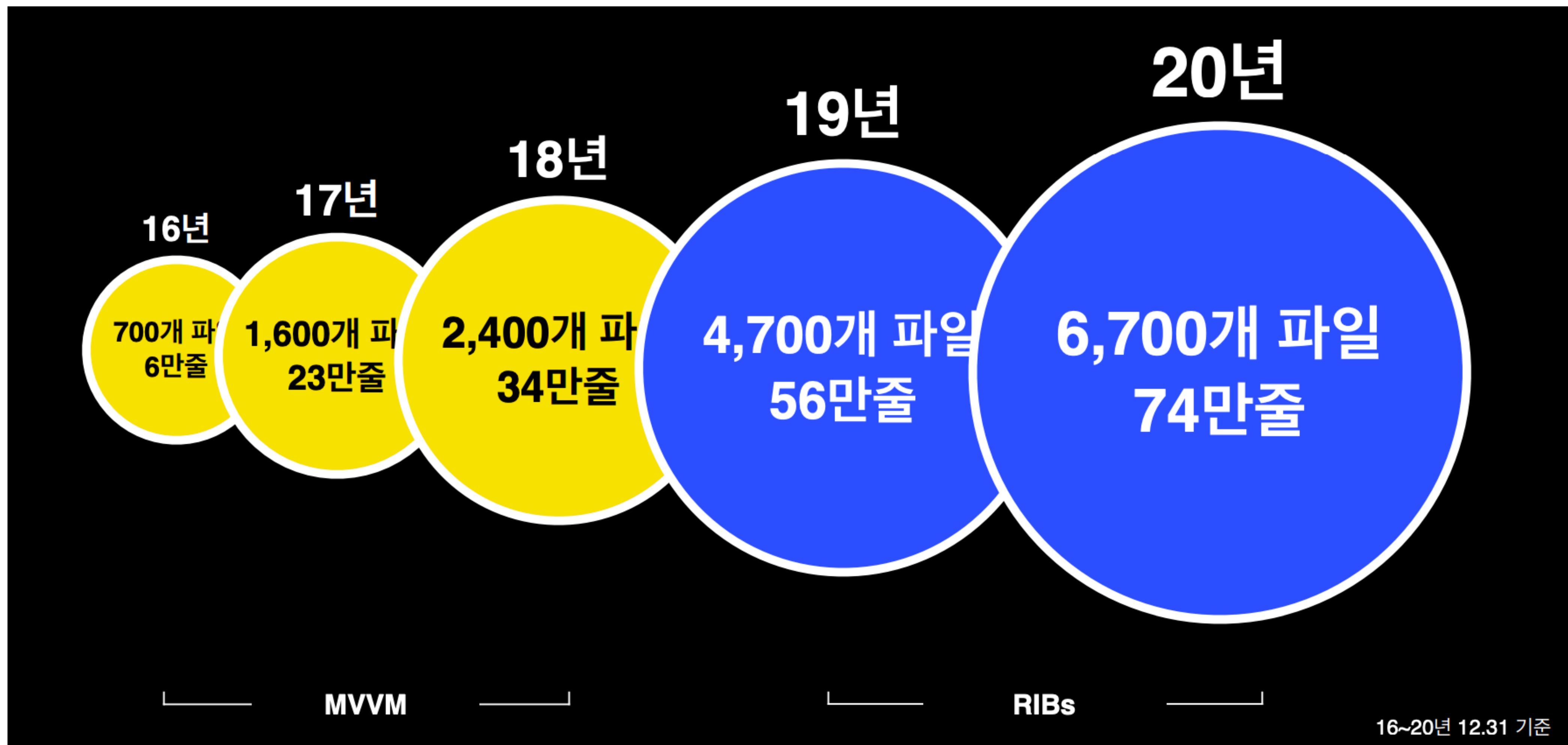
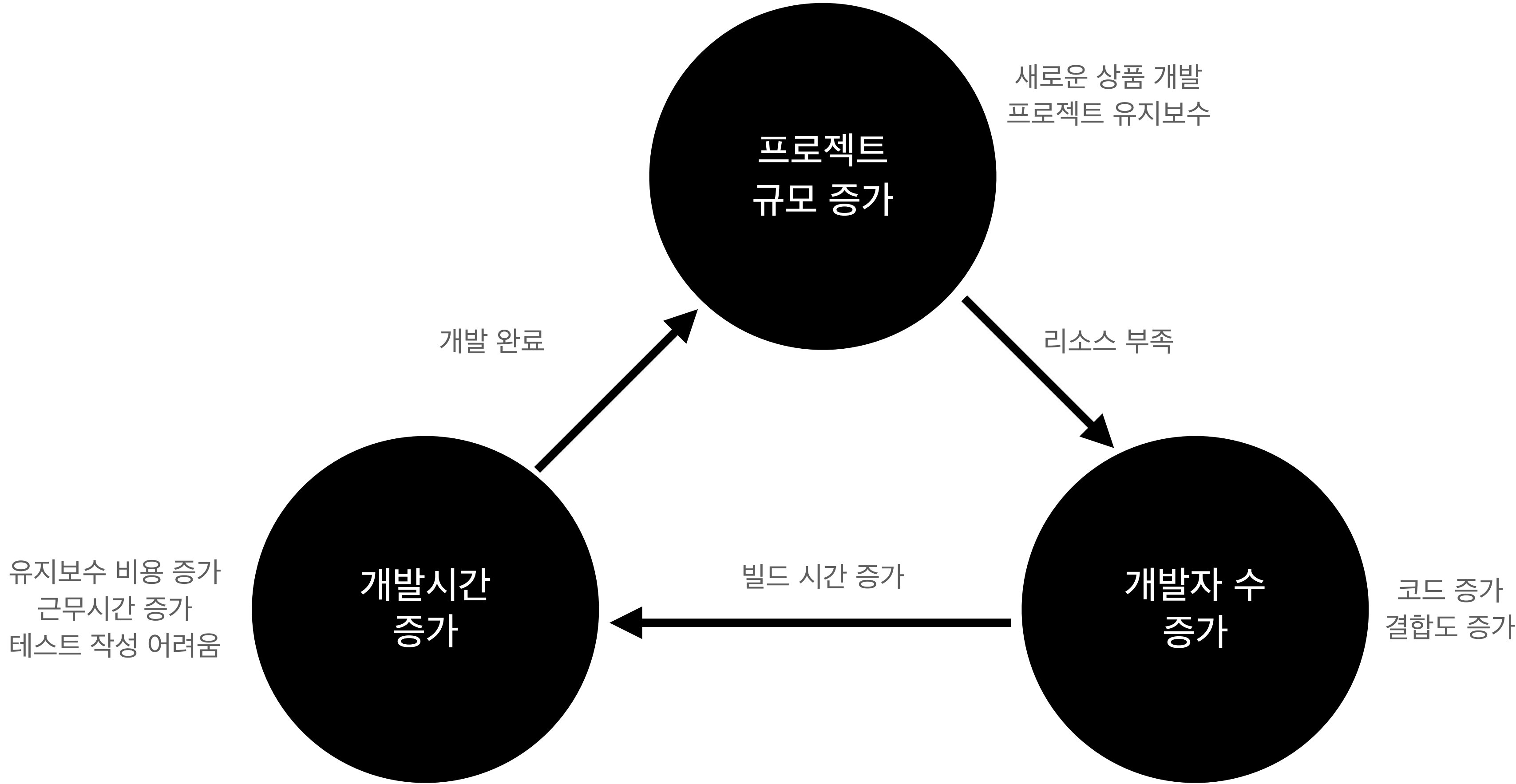## Project.phxproj

- 각 파일들의 참조값들을 UUID들로 정의

Workspace

- Xcode의 Project 및 기타 리소스를 그룹화하여 합께 작업할 수 있는 Xcode document

- 다수의 Project를 사용하고 싶은 경우, workspace 하위로 관리

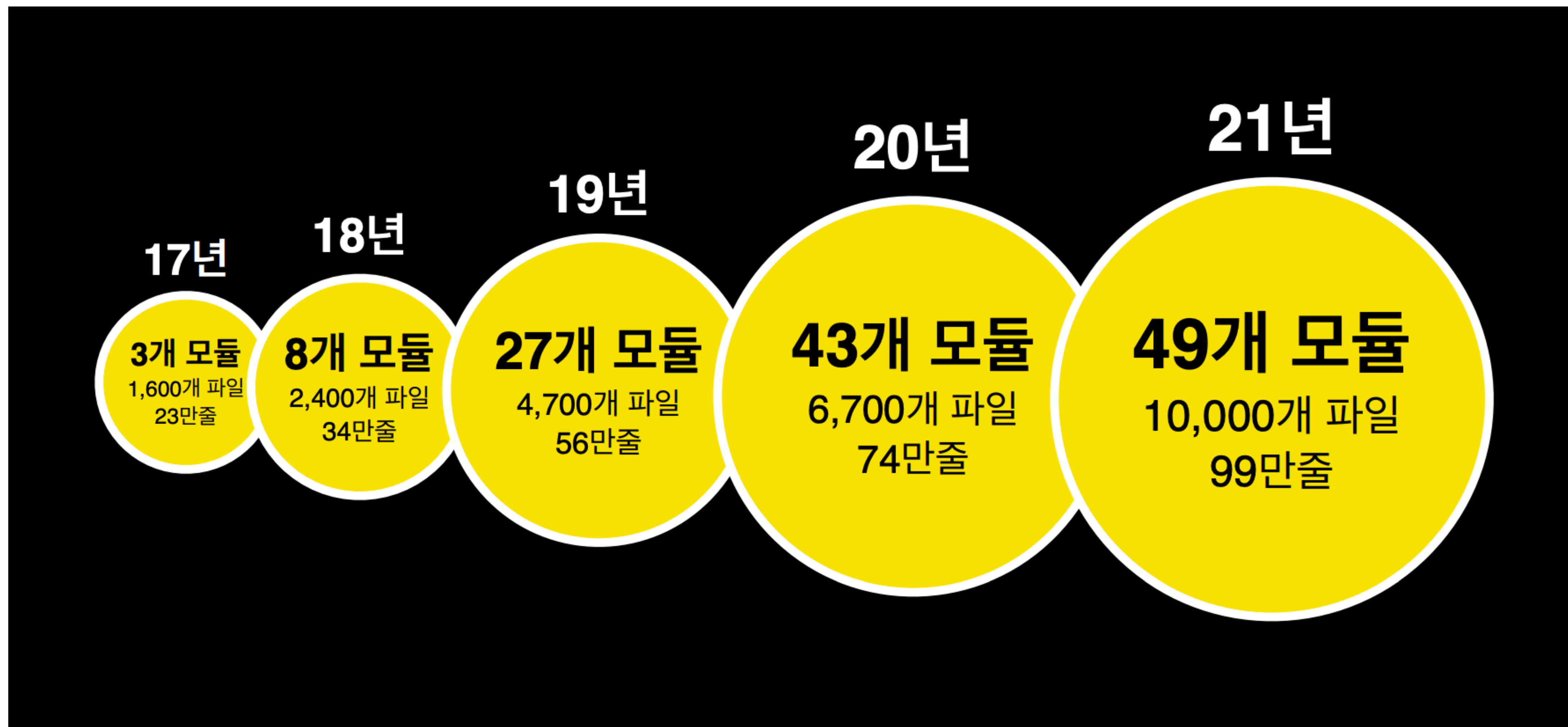- 각 Project의 파일을 구성하는 것 외에도 workspace에 포함된 Project들과 Target간의 관계를 제공
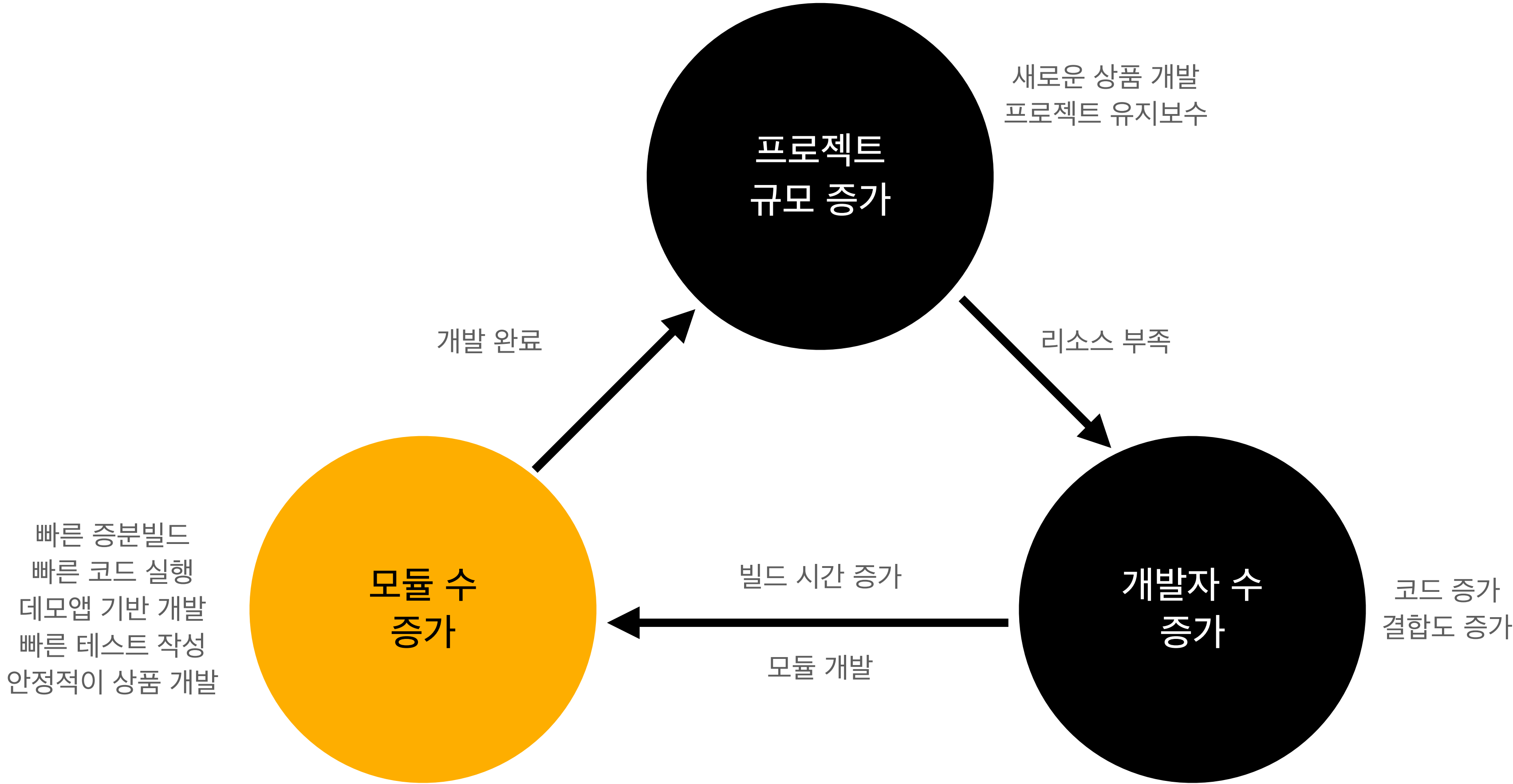
모듈화

개발 프로세스

🧐

모듈화로 해결할수 있지 않을까?

모듈화

17년
3개 모듈
1,600개 파일
23만줄

18년
8개 모듈
2,400개 파일
34만줄

19년
27개 모듈
4,700개 파일
56만줄

20년
43개 모듈
6,700개 파일
74만줄

21년
49개 모듈
10,000개 파일
99만줄

모듈화

프로젝트
규모 증가

새로운 상품 개발
프로젝트 유지보수

개발 완료

리소스 부족

모듈 수
증가

빠른 증분빌드
빠른 코드 실행
데모앱 기반 개발
빠른 테스트 작성
안정적이 상품 개발

빌드 시간 증가

개발자 수
증가

코드 증가
결합도 증가

모듈 개발

모듈 조립

안정적인 기능, 상품 개발

빠른 빌드

UI, 비즈니스 로직 개발 및 실행

모듈화
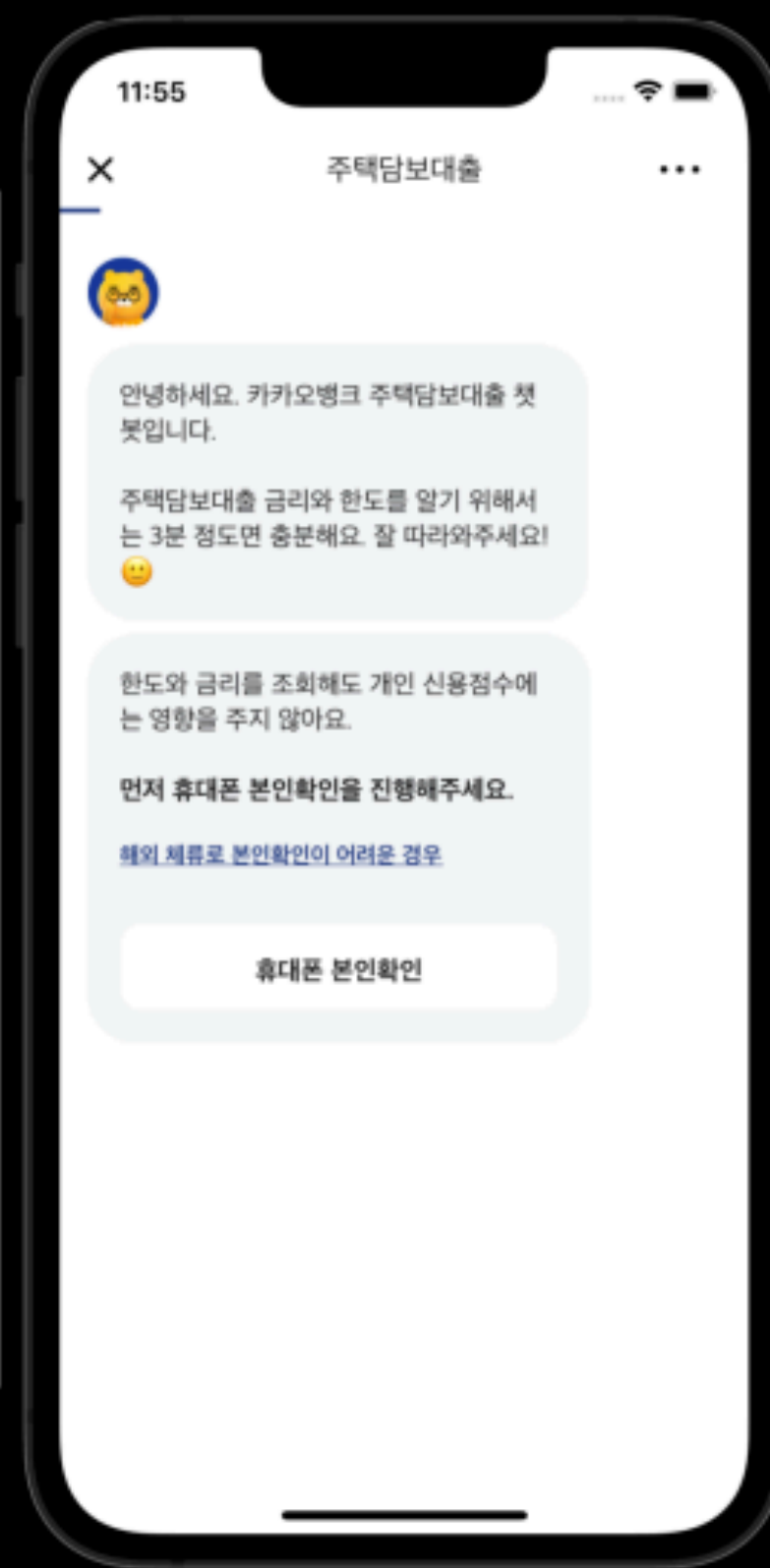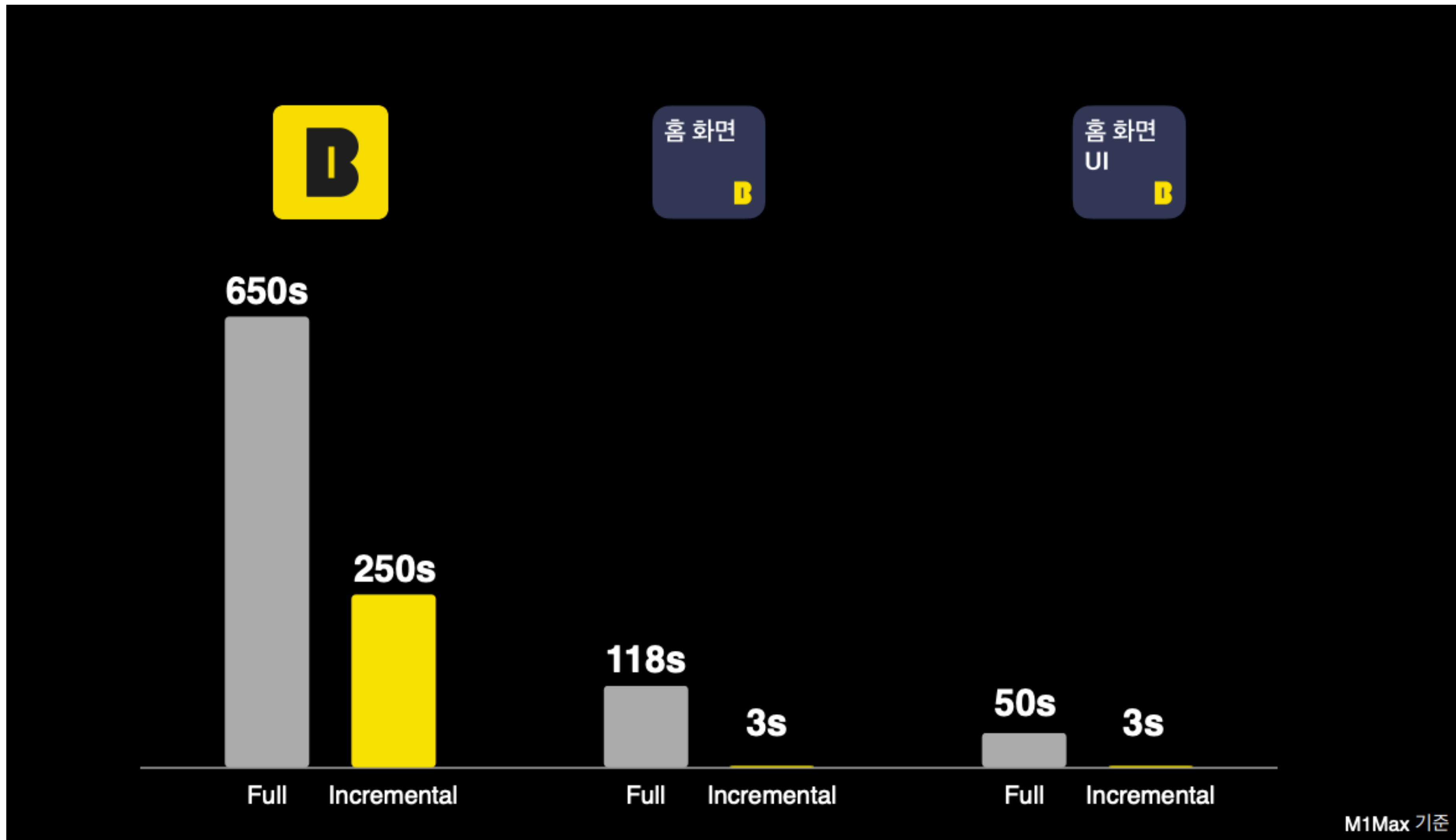
모듈화

Tuist 사용해 보기

Install

The first thing that we need to do to get started is installing the tool. To do so, you can run the following commands in your terminal:

```
curl -Ls https://install.tuist.io | bash
```

```
tuist init --platform ios
```

The `init` command will bootstrap an iOS application, which includes the `Info.plist` files, an `AppDelegate.swift`, a tests file, and a `Project.swift` that contains the definition of the project.

> If you have used the Swift Package Manager before, the `Project.swift` file is the equivalent to the `Package.swift`.

ℹ️ **SWIFTUI TEMPLATE**

`tuist init --platform ios --template swiftui` will bootstrap a SwiftUI iOS project instead.

Tuist 사용해 보기

```
tuist edit
```

```swift
import ProjectDescription

let project = Project(
    name: "MyApp",
    organizationName: "MyOrg",
    targets: [
        Target(
            name: "MyApp",
            platform: .iOS,
            product: .app,
            bundleId: "io.tuist.MyApp",
            infoPlist: "Info.plist",
            sources: ["Sources/**"],
            resources: ["Resources/**"],
            headers: .headers(
                public: ["Sources/public/A/**", "Sources/public/B/**"],
                private: "Sources/private/**",
                project: ["Sources/project/A/**", "Sources/project/B/**"]
            ),
            dependencies: [
                /* Target dependencies can be defined here */
                /* .framework(path: "framework") */
            ]
        ),
        Target(
            name: "MyAppTests",
            platform: .iOS,
            product: .unitTests,
            bundleId: "io.tuist.MyAppTests",
            infoPlist: "Info.plist",
            sources: ["Tests/**"],
            dependencies: [
                .target(name: "MyApp")
            ]
        )
    ]
)
```

터미널에서 ctrl + c 입력하여 편집 종료

Tuist 사용해 보기

```
tuist generate
```

우리가 알던 프로젝트가 나옴
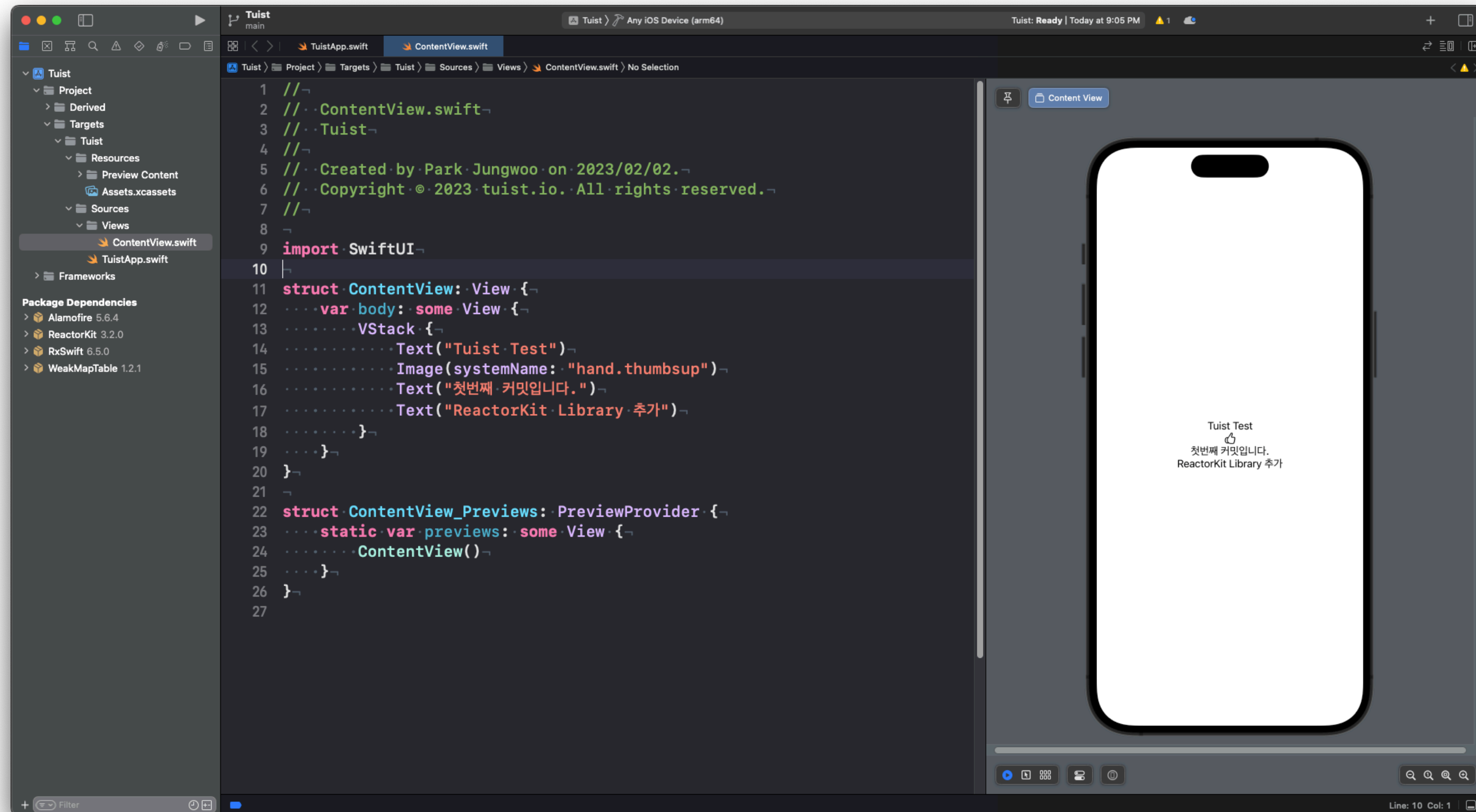
Tuist 사용해 보기
중요한 점

Git push

항상 tuist generate

Pbxproj conflict 해방

모듈화의 이점

- **코드 안전성**

  - 모듈간 결합도가 낮아지고 응집도가 높아지는 만큼 안정성

- **개발 생산성**

  - 모듈별 의존성이 낮아지면 고려해야하는 범위가 낮아져 개발속도 향상

  - 코드 리뷰 시 관련 모듈만 검토 가능

- **빌드 속도**

  - 자주 변경되지 않는 모듈의 빌드가 발생하지 않아 빌드 속도 증가

- **사용 목적에 맞는 코드 배치**

  - 용도와 목적에 맞는 코드 배치로 이해하기 쉬운 프로젝트 구성

참고

- https://www.youtube.com/watch?v=7rvwGLhor-I&t=173s

- https://ios-development.tistory.com/1004

- https://leeari95.tistory.com/74#Tuist%EB%A5%BC-%EC%82%AC%EC%9A%A9%ED%95%98%EA%B8%B0-%EC%A0%84%EC%97%90-%EC%95%8C%EC%95%84%EC%95%BC-%ED%95%A0-%EA%B0%9C%EB%85%90

- https://docs.tuist.io/tutorial/get-started/

- https://www.youtube.com/watch?v=Up0wYoyY3o0&t=1497s