Clean Architecture

R.C.Martin

경계는 언제 필요한가?

경계는 항상 존재하지만, 언제 필요한지를 생각하자

경계를 구현하는 cost는 만만치 않다

경계가 무시되었다면 오히려 cost가 증가할 수도 있다

우리는 어떻게 해야할까?

You Aren't Going to Need It

Overengineering is worse than underengineering

Tracking

Main Component

가장 낮은 수준의 Policy

지저분한 컴포넌트

메인은 플러그인, 경계 바깥에 위치

SOA(Service Oriented Arch)

서비스는 프로세스, 플랫폼 경계를 가로지르는 함수 호출

결합 분리 오류 / 개발 및 배포 독립성의 오류

Cross-cutting Concern

아키텍쳐 경계는 서비스 사이에 있지 않다

테스트 경계

테스트는 시스템 외부에 있지 않다 => CI에서 돌려봐도 그렇지 않은가?

시스템에 결합된 테스트는 좋지 않다

Fragile Test is bad

세부사항

야, 아무튼 비즈니스 로직 빼면 다 세부 사항이야 => ~~WTF?~~

Framework와 결혼하지 마라

선택적으로 결정할 수 있어야 한다

BUT 신입 중 기혼자가 있다면 적응하기 어려울 수도

고민해봐야 하는 요소라고 생각

실용주의적으로 행해라

가능하다면 선택사항을 열어두되, 실용주의적으로 행하라

팀의 규모, 기술 수준, 해결책의 복잡성을 일정, 예산 제약과 동시에 고려하라

마지막으로

Uncle Bob 대단하다

클린 아키텍쳐는 종교가 아니다

실용주의는 어떤 특정 기술에 매이면 안되며,

개별 상황마다 그 상황에서 좋은 해결방안을 고를 수 있어야 한다.

그를 위한,

경험과 배경지식이 필요하다.

- 실용주의 프로그래머 일부 발췌