



Component

배포 단위

독립적 배포, 개발이 가능한.

Component Plugin Arch

링킹 로더

More bigger Program and..

Moore is winner

REP: 재사용/릴리즈 등가 원칙

재사용 단위는 릴리즈 단위와 같다.

Cohesive Component

CCP: 공통 폐쇄 원칙

동일 이유, 동일 시점 변경 => 묶어!

다른 이유, 다른 시점 변경 => 분리!

CCP: 공통 폐쇄 원칙

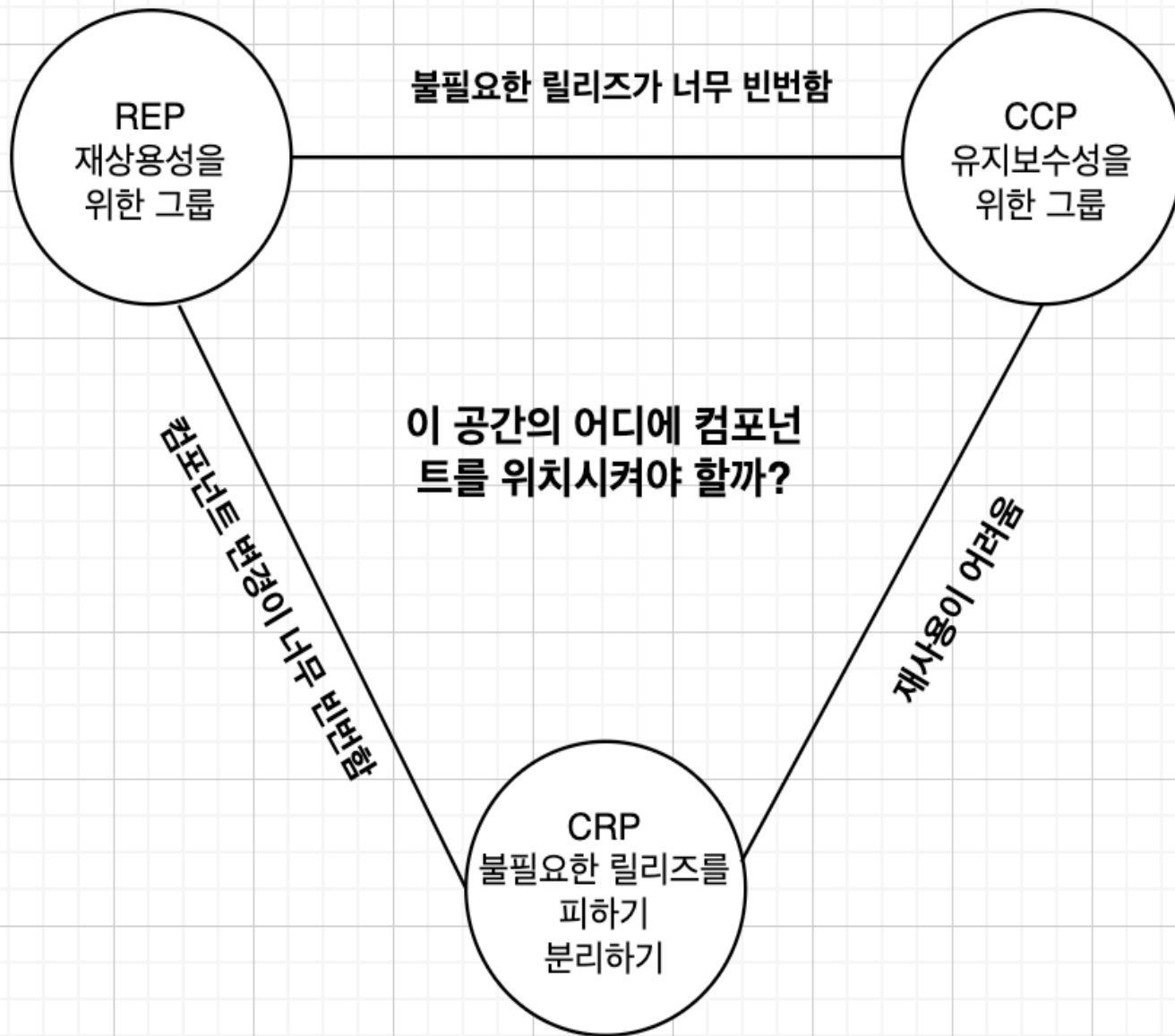
컴포넌트 수준의 SRP
유지보수성 > 재사용성?
OCP

CRP: 공통 재사용 원칙

클래스 사이의 의존성에 의한 배치

강하게 결합되지 않은 클래스 **NOT IN** Same Component

ISP, 필요하지 않은 것에 의존하지 마라



아키텍트의 판단
보통 초기 => 재사용성 희생



시간이 흐를 수록
개발 팀이 주의를
기울이는 부분이 변함

DAG : Directed Acyclic Graph(비순환 방향 그래프)

ADP : 의존성 비순환 원칙

컴포넌트 의존성 그래프에 cycle이 있어서는 안된다.

Fxxk Monring after syndrome

어떤 컴포넌트를 따라가도 최초의 컴포넌트로 못 돌아간다

순환 끊기

1. DIP

2. New Component (feat. Jitters)

SDP : 안정된 의존성 원칙

안정성의 방향으로(더 안정된 쪽에) 의존하라
변동하는 시스템
Stability?

안정성 지표

Fan-in / Fan-out / I

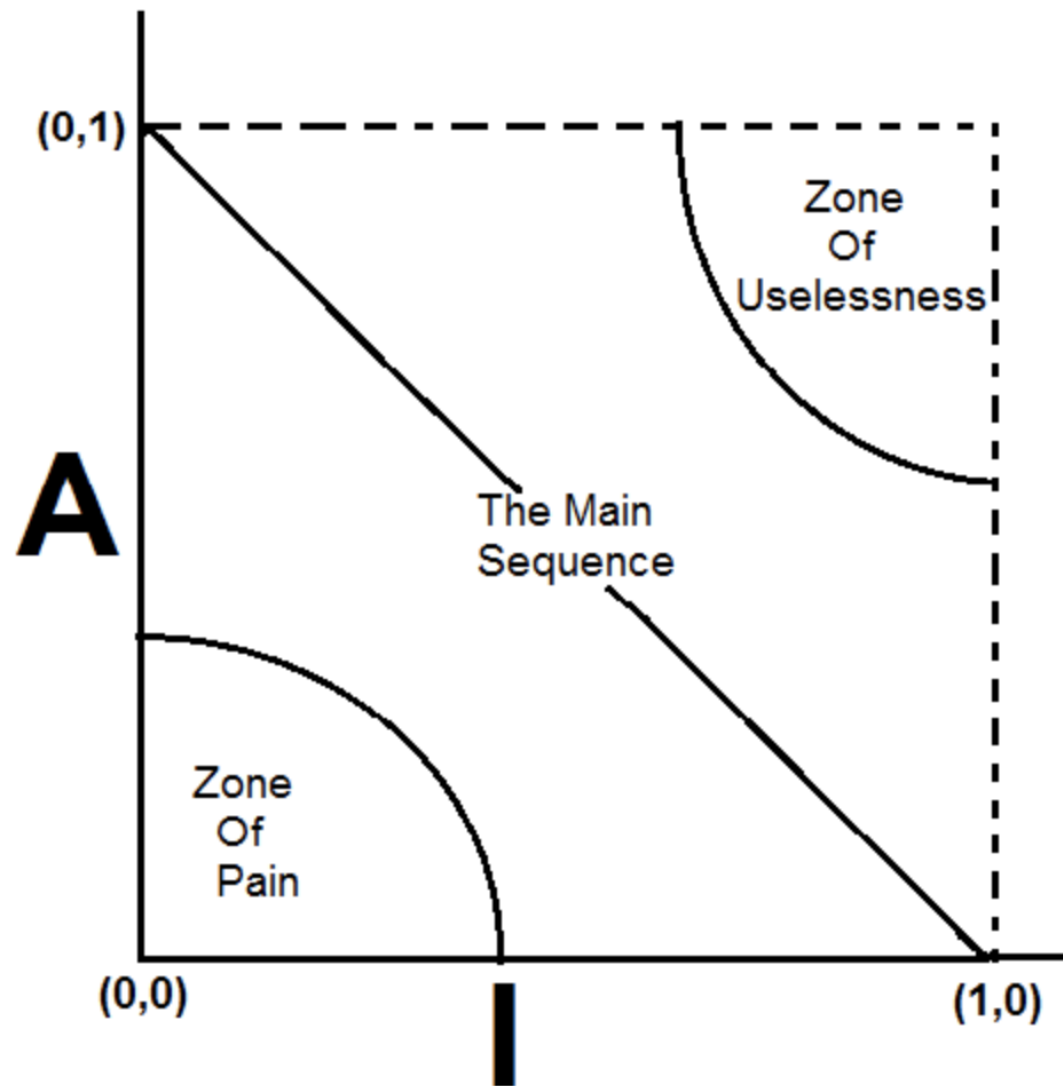
의존성 방향으로 갈수록 I가 감소

SAP : 안정된 추상화 정책

컴포넌트는 안정된 정도만큼만 추상화되어야 한다.

안정된 추상화 원칙

안정된 컴포넌트 = 인터페이스, 추상 클래스 for 확장



The I/A graph

아키텍처

시스템의 생명주기를 지원
프로그래머의 생산성을 최대화

개발 / 배포 / 운영 / 유지보수

Spelunking (탐사)

Policy & Detail

선택사항 열어두기

좋은 아키텍트는 결정되지 않은 사항의 수를 최대화한다.

장치 독립성

독립성

Usecase / Operation / Development / Deployment

시스템의 의도

Service Oriented Architecture

중복



결합 분리 모드

소스 수준 분리 모드

배포 수준 분리 모드

서비스 수준 분리 모드

Façade : 불어에서 온 단어로, 건물의 정면을 뜻함 BUT SW에서는 파사드 패턴을 주로 의미

경계 해부학

다양한 경계들

로컬 경계 & 지연을 중요하게 고려해야 하는 경계

