

# #7 Optimized and Balanced SW Development Process

## 10팀

202112349 조용국

202211342 이윤원

202211359 장상욱

202211269 김민수

# Table of Contents

Overview	3
Inception Phase	5
OOA	8
OOD	11
OOI	14
CI/CD	17
결론	20

# Overview

# Overview

### Inception

<b>AI</b> 요구사항 후보 추출 Actor/FR/NFR 분류 Use Case 후보 작성	<b>인간</b> 요구사항 후보 추출 Actor/FR/NFR 분류 Use Case 후보 작성
--	--

**인간**

검토 승인 책임

### OOA

<b>AI</b> Use Case 초안 작성 도메인 개념/모델 초안 UC-FR 일관성 검증	<b>인간</b> 승인 및 책임, UC 확정 시스템 바운더리 확정 도메인 Object 확정
---	---

### OOD

<b>AI</b> CD, SD 초안 아키텍처 제안 설계 대안 및 리스크 측정	<b>인간</b> 아키텍처 클래스 책임 결정 설계 검토 및 승인 품질 책임
---	--

### OOI

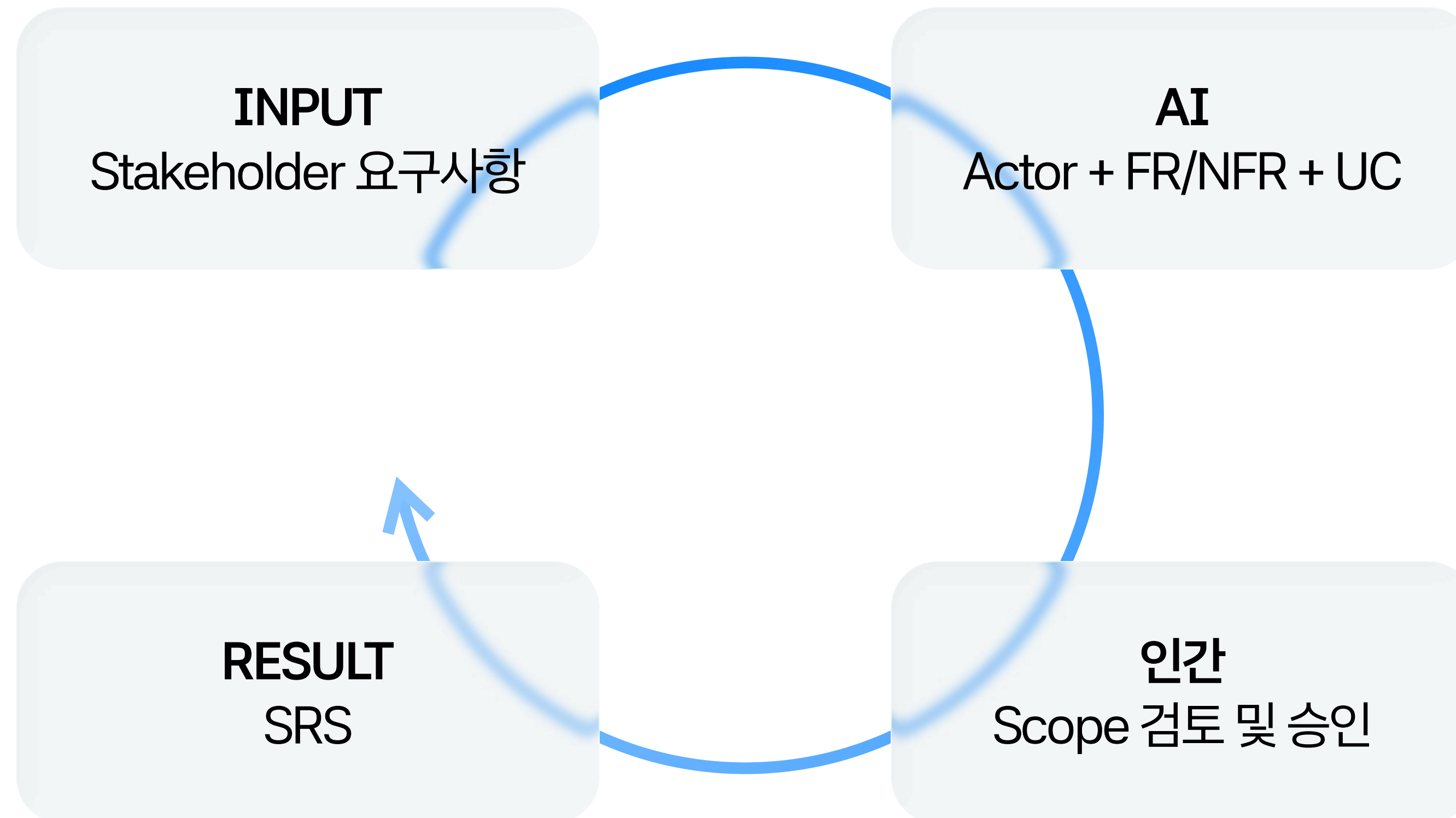
<b>AI</b> 코드 스켈레톤 및 구현 지원 테스트 결과 분석 리팩토링/문서화 지원	<b>인간</b> 코드 리뷰 및 수정 테스트 케이스 작성 및 결과 검토 보안/성능 검토
--	---

### CI/CD

<b>AI</b> 빌드/테스트/운영 로그 분석 장애 분석/배포 체크리스트 /릴리즈 노트 작성	<b>인간</b> 배포 승인 및 판단 롤백/핫픽스 결정 운영 리스크 수용 판단
--	--

# Inception Phase

# Inception Phase



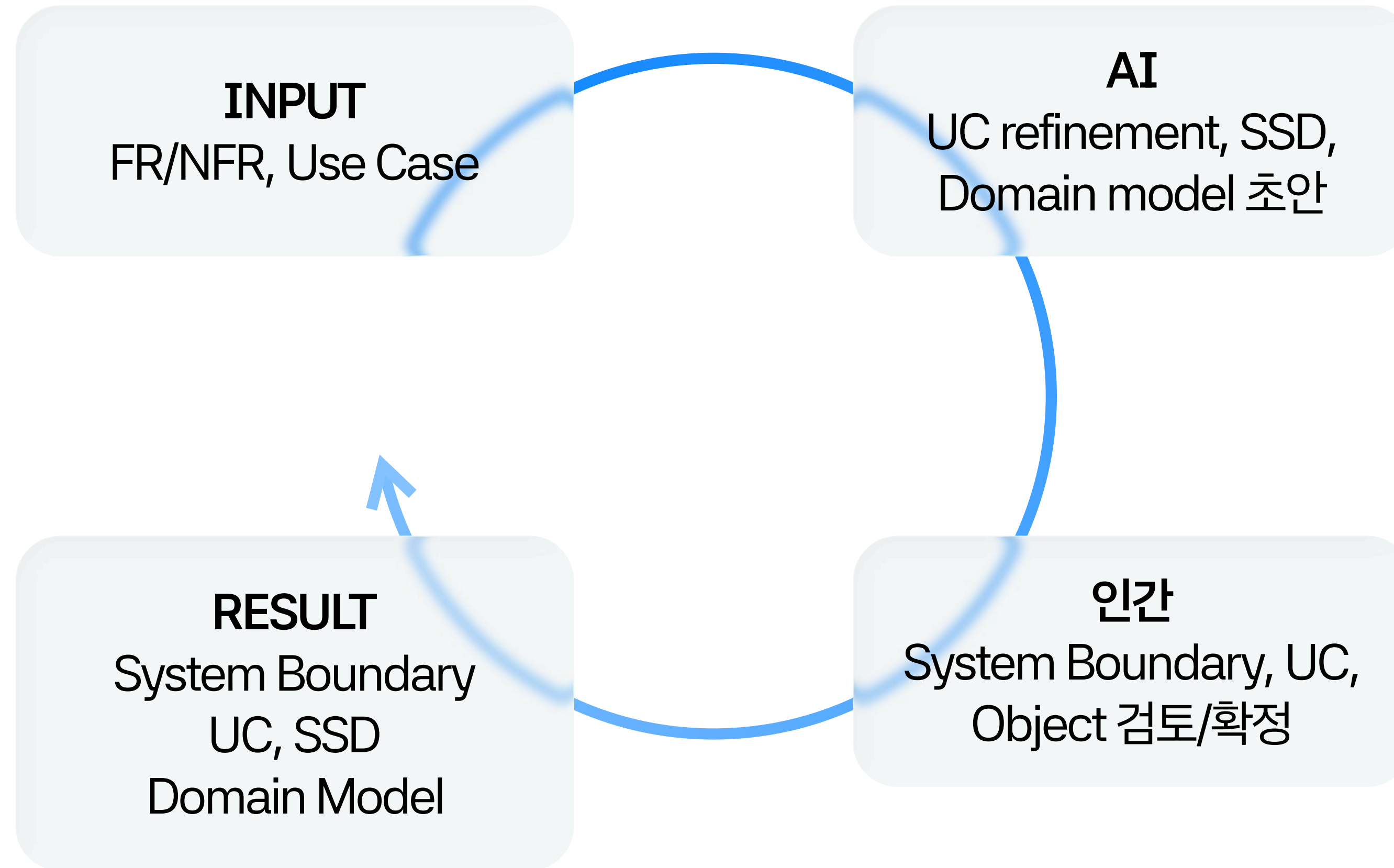
# Inception Phase

<b>한계점</b>	<p>stakeholder의 직접적인 의도 파악이 어려움</p> <p>실제 stakeholder가 요구하지 않은 사항을 암묵적으로 추가할 수 있음</p> <p>조직 내 컨벤션에 대한 이해가 떨어짐</p> <p>측정 불가능한 / 모호한 기준에 대한 판단력이 떨어짐</p>
------------	--

누가	무엇을	왜	어떻게&어디까지
<b>인간 책임</b>	<p>Stakeholder 인터뷰</p> <p>System Boundary 조정</p> <p>FR/NFR 최종 결정</p> <p>AI 산출물 검토 및 승인</p>	<p>Requirements는 product/domain 중심</p> <p>인간이 책임을 져야함</p> <p>인간이 최종 결정권을 가짐</p>	<p>인터뷰 및 회의로 근거 수집</p> <p>→ 범위, 제약, 제외사항 결정</p> <p>→ AI 초안 리뷰</p> <p>→ 승인 혹은 반려 및 수정 요청</p>
<b>AI Agent 서포트</b>	<p>Actor/FR/NFR 후보 분류</p> <p>Requirements 구조화</p> <p>UC 초안 작성</p> <p>UC-FR 일관성 검증</p> <p>문서화</p>	<p>반복 작업 수행에 강함</p> <p>누락, 충돌 등 문제를 조기에 발견</p>	<p>인터뷰 내용 주입</p> <p>→ Actor/FR/NFR 후보 도출 및 분류</p> <p>→ UC-FR 연결</p> <p>→ 모호성 파악</p> <p>→ 반복</p>

OOA

# OOA



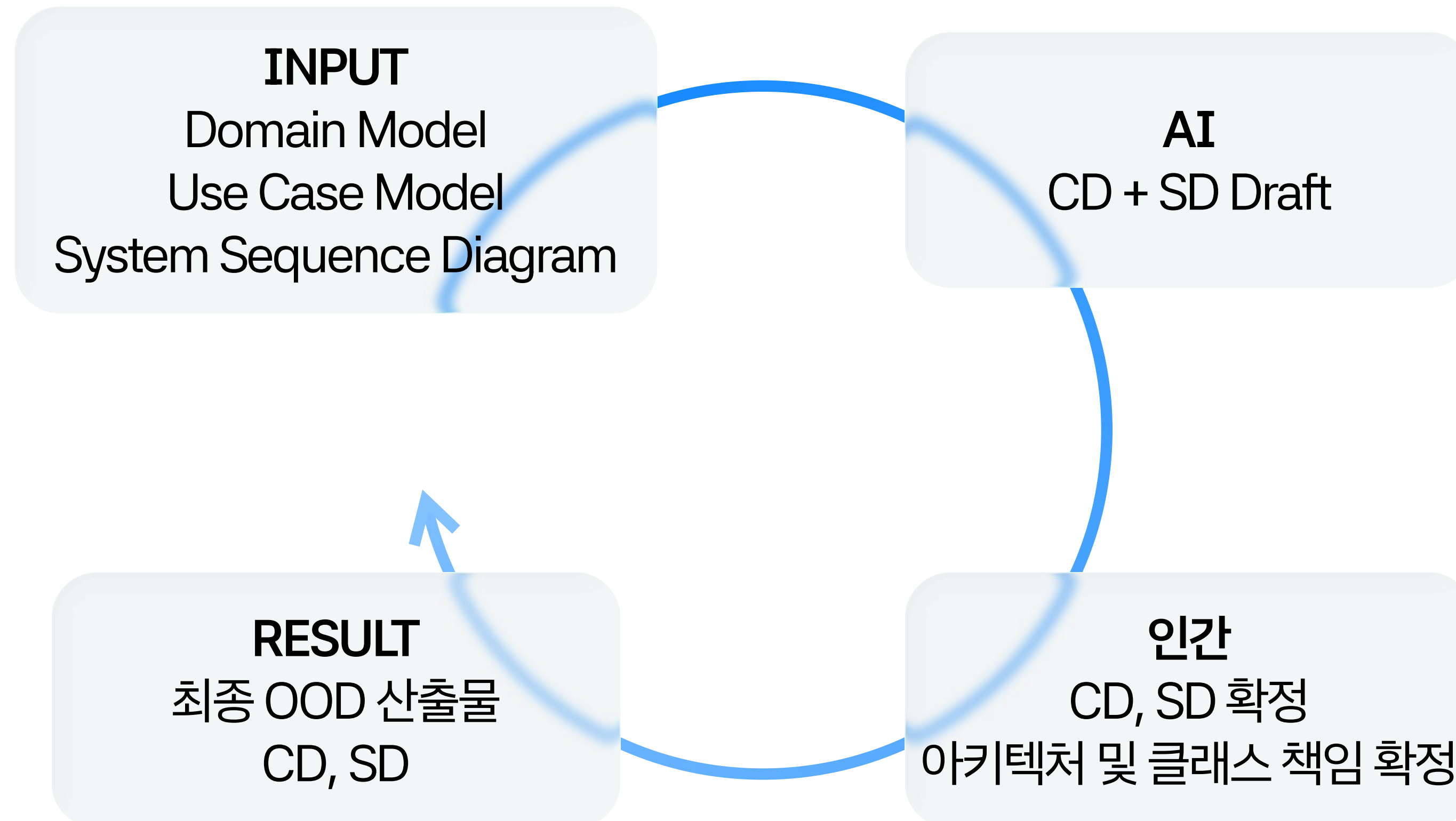
# OOA

<b>한계점</b>	<p>사람과 AI가 생각하고 있는게 일치하는지 알 수 없음</p> <p>AI가 시스템에 대해 정확히 이해하고 있는지 모름</p>
------------	---

누가	무엇을	왜	어떻게&어디까지
<b>인간</b> 책임	<p>System Boundary 확정</p> <p>AI 산출물 검토 및 승인</p>	<p>AI가 생각하는 System Boundary와 인간의 생각이 다를 수 있기에 최종은 인간이 결정</p>	<p>AI에서 나온 UC refinement, SSD 초안 검토 &amp; 수정하고 System Boundary 확정</p> <p>초안 Domain Model에서 인간이 최종 Domain Model 확정</p>
<b>AI Agent</b> 서포트	<p>UC refinement 초안 작성</p> <p>UC-FR 일관성 검증</p> <p>SSD 초안 생성</p> <p>domain model 초안 생성</p>	<p>반복 작업 수행에 강함</p> <p>일관성 문제를 조기에 발견</p> <p>객체 도출에 아이디어 제공</p>	<p>UC refinement 초안 작성 → UC-FR 일관성 검증</p> <p>Use Case를 통해 SSD, Domain 초안 작성</p>

OOD

# OOD



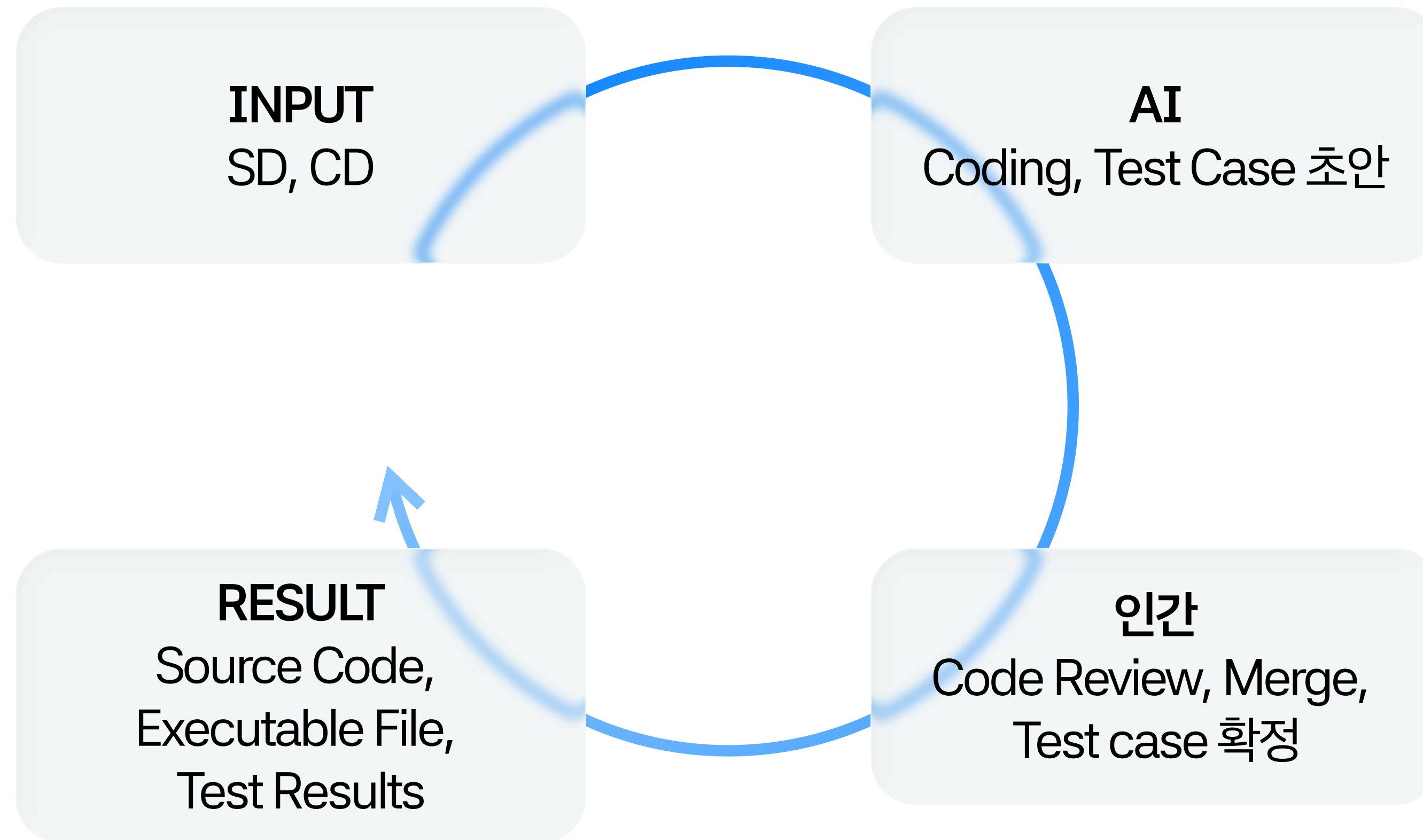
# OOD

한계점	요구사항에 적절한 아키텍처를 설계하기에 부족하다 유지/보수 단계를 고려한 아키텍처 및 클래스 책임을 설계하지 못한다
-----	---

누가	무엇을	왜	어떻게&어디까지
인간 책임	CD, SD 최종 확정 아키텍처 설계 확정 클래스 책임 확정	유지 및 보수 단계를 고려한 SOLID 원칙을 준수하는 아키텍처를 설계해야 할 필요성	초안 기준으로 FR 검증 및 유지/보수 단계를 고려한 Class Diagram 및 Sequence Diagram 확정, 아키텍처 확정
AI Agent 서포트	Class Diagram 초안 작성 Sequence Diagram 초안 작성	빠른 초안 작성을 통해 업무 효율 극대화	이전 단계의 Use Case Model, System Sequence Diagram, Domain Model 주입 → Class Diagram 및 Sequence Diagram 초안 작성

OOI

# OOI



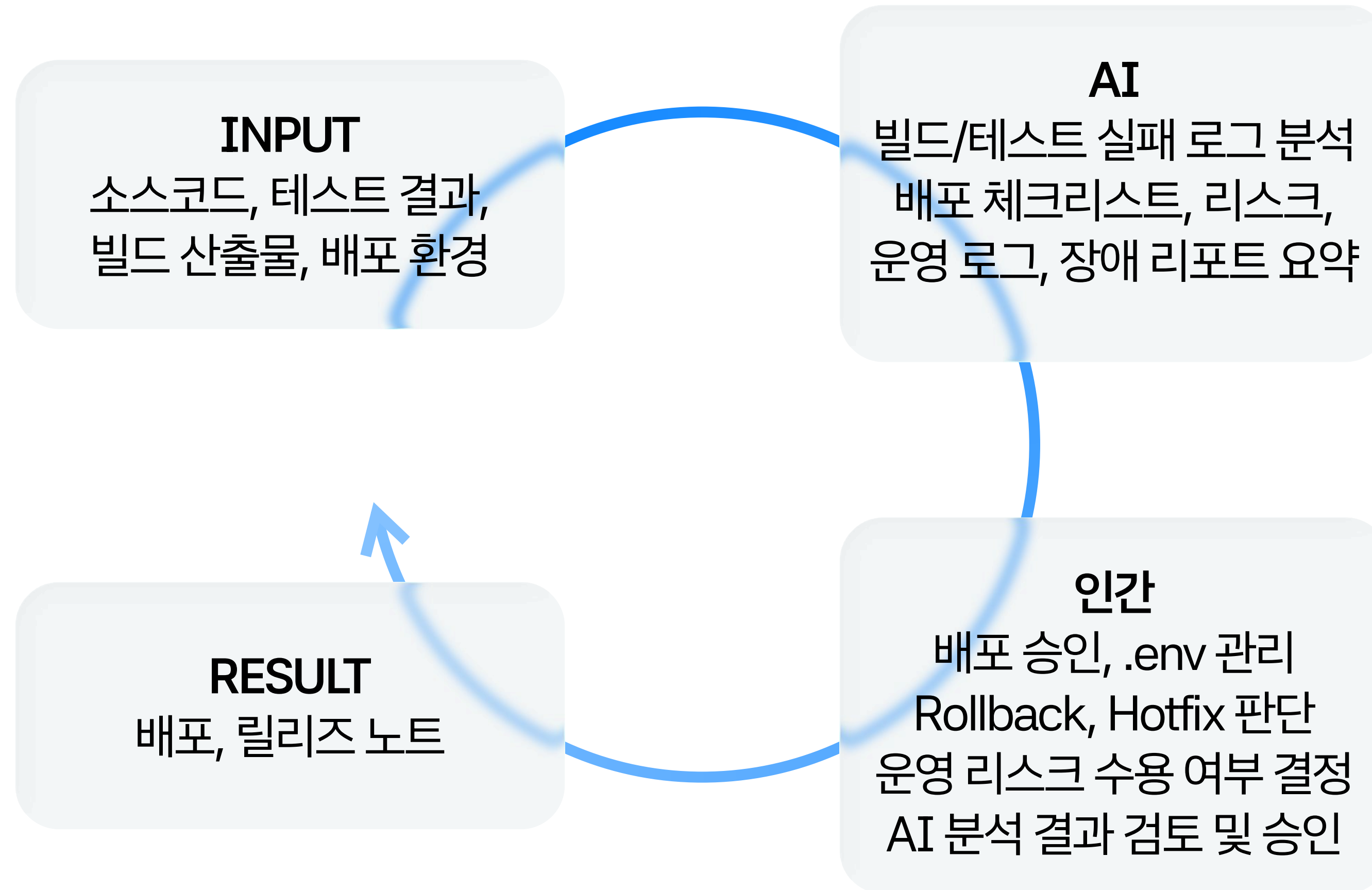
# OoI

<p><b>한계점</b></p>	<p>성능 / 최적화를 고려하지 않은 코드          잘못된 merge로 인한 문제 상황 발생          요구사항을 만족시키지 않고도 완료 보고          충분하지 못한 Test Case</p>
-------------------	---

누가	무엇을	왜	어떻게&어디까지
<p><b>인간 책임</b></p>	<p>코드 리뷰          merge          Test - Edge, Corner Cases          Test - 결과 분석</p>	<p>AI가 생성한 코드는 취약점, 성능을 보장할 수 없음.          Merge는 잘못 실행하면 배포까지도 문제 발생 가능성          Test에서 시스템의 안정성을 보증하는 Edge Case까지 AI에게 맡기기에는 부족</p>	<p>AI가 생성한 코드를 비즈니스 로직 검증, 사이드 이펙트 및 취약점 확인, 성능 측정을 통해 코드의 완성도를 점검          AI가 생성한 Test Case 초안 검토 및 Edge, Corner Case 작성          Test 결과 해석 및 Coverage 검토</p>
<p><b>AI Agent 서포트</b></p>	<p>SD, CD 기반 코드 생성          Test Case 초안 작성          Unit/System Test 실행          문서화</p>	<p>반복 작업          방대한 양의 코드 학습          빠른 구현</p>	<p>SD, CD를 기반으로 코드 작성          Test Case 초안 작성          검토된 Test 실행, 결과 요약</p>

CI/CD

# CI/CD



# CI/CD

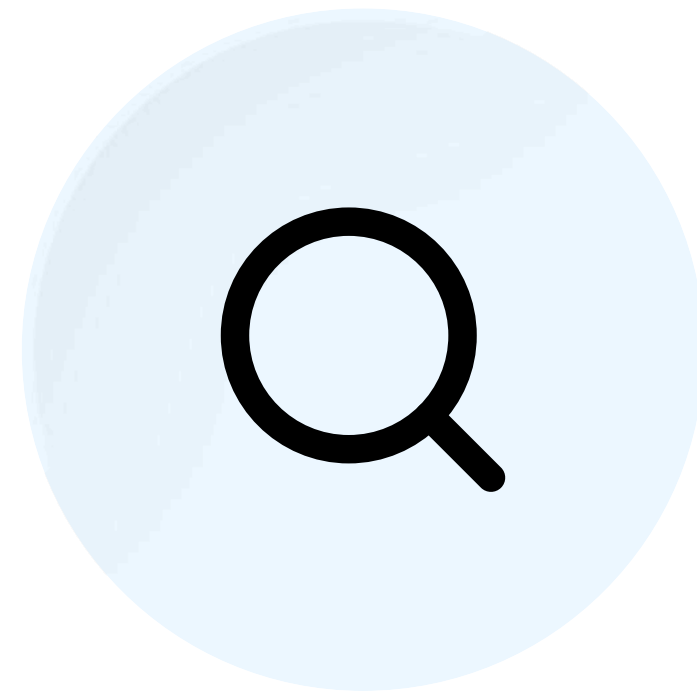
<b>한계점</b>	<p>배포 환경 설정, Secret, Rollback 판단은 잘못될 경우 실제 운영 장애로 이어질 수 있음</p> <p>AI는 빌드/테스트 실패 로그를 분석할 수 있지만, 운영 리스크를 최종 판단하기는 어려움</p> <p>AI가 생성한 CI/CD 설정은 프로젝트 컨벤션, 인프라 구조, 보안 정책을 완전히 이해하지 못할 수 있음</p> <p>운영 로그 분석 결과가 원인 후보일 뿐, 실제 원인이라고 단정할 수 없음</p>
------------	---

누가	무엇을	왜	어떻게&어디까지
<b>인간 책임</b>	<p>배포 승인</p> <p>Rollback / Hotfix 판단</p> <p>환경 변수·Secret 검토</p> <p>운영 리스크 수용 여부 결정</p> <p>AI 산출물 검토 및 승인</p>	<p>CI/CD는 실제 운영 환경에 직접 영향을 주는 단계임</p> <p>잘못된 배포는 서비스 장애와 데이터/보안 문제로 이어질 수 있으므로 최종 판단과 책임은 인간이 가져야 함.</p>	<p>AI가 생성한 배포 체크리스트와 로그 분석 결과를 검토</p> <p>→ 환경 설정, Secret 노출 여부, 테스트 통과 여부, 영향 범위 확인</p> <p>→ 배포 승인 또는 반려</p> <p>→ 장애 발생 시 Rollback/Hotfix 여부 결정</p>
<b>AI Agent 서포트</b>	<p>빌드/테스트 실패 로그 분석</p> <p>릴리즈 노트 초안 작성</p> <p>배포 체크리스트 생성</p> <p>운영 로그/장애 리포트 요약</p>	<p>반복적인 로그 분석, 문서화, 체크리스트 생성에 강함</p> <p>실패 원인 후보를 빠르게 좁히고 운영 피드백을 개발 단계로 환류시키는 데 도움을 줄 수 있음</p>	<p>Source Code, Test Results, Pipeline Log, Deployment Config 입력</p> <p>→ 실패 원인 후보 분석</p> <p>→ Release Note / Deploy Checklist 작성</p> <p>→ 운영 로그 요약</p> <p>→ Incident Report</p>

# 결론

# 결론

앞만 바라볼 뿐, 숲을 바라보지 못한다



검토



승인



책임