

# 객체지향개발방법론

# Optimized and Balanced SW Development Process

---

4팀 김승찬, 유호재, 윤현서, 최서진

# INDEX

목차페이지

- 01 기본사항
- 02 큰 흐름
- 03 Inception
- 04 Elaboration - OOA
- 05 Elaboration - OOD
- 06 Elaboration - OOI
- 07 Transition
- 08 Operation
- 09 큰 흐름
- 10 제안

## ✓ 기본 사항

# AI의 작업 결과물을 사람이 검토한다.

방식	설명
기준과 비교	(요구 사항, 제약 조건, 이전 산출물, 평가 기준) 과 비교해서 판단한다.
오류, 누락, 과잉, 모순 발견	틀리거나 누락된 내용, 요구하지 않은 내용의 임의 추가, 이전 결정, 다른 산출물과 충돌을 찾는다.
AI의 가정과 확정을 구분	AI가 애매한 부분을 스스로 확정해서 답하는 경우를 구분해야한다.
반례와 실패 상황 찾기	AI 산출물이 옳아 보여도 언제 실패하는지 반례를 확인해야 한다.
검증 가능한 것은 직접 검증	모든 산출물이 실행 가능한 것은 아니지만 검증 가능한 부분은 말로만 판단하지 않는다.

## ✓ 기본 사항

# 추가 설명 - 검증 가능한 것은 직접 검증

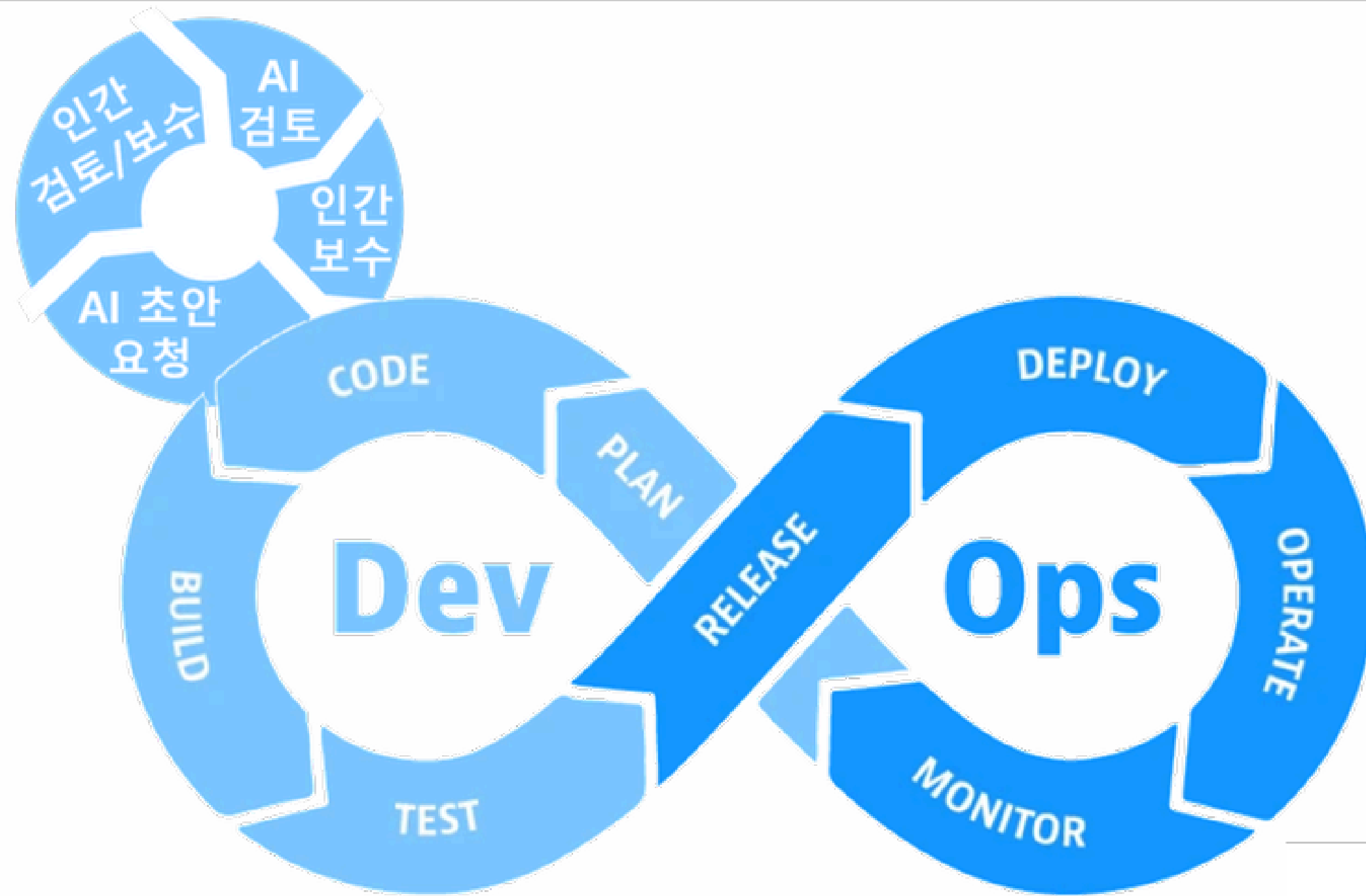
사람이 모든 AI 산출물을 끝까지 읽으며 판단하는 것은

**비효율적이며  
사람조차 틀릴 수 있다.**

실행, 테스트, 스크립트, 도구로 확인 가능한 부분은 **자동화된 방식으로 검증한다.**

즉, AI 결과물을 검토할 때 사람의 역할은 **모든 내용을 직접 읽는 사람이 아니라,  
무엇을 자동 검증할 수 있는지 정하고, 검증 결과를 해석하는 사람에 가깝다.**

# ✓ 큰 흐름



## ✓ Inception



**“사람이 판단할 수 있도록  
초안, 대안, 위험, 질문을 빠르게 만들어주는 보조자”**

프로젝트 방향을 대신 결정하지 않는다.

사람이 판단할 수 있도록 초안을 빠르게 만들고, 가능한 대안과 위험 요소 제시

아직 결정되지 않은 질문을 정리하여 초기에 중요한 사항을 놓치지 않도록 돕는다

# ✓ Inception



## AI Agent 활용 예시

### 1. AI Agent에게 프로젝트 방향 제시

“3주동안 5명이 Python 으로 캠퍼스 배경 게임을 만들고, 발표와 보고서 평가를 받는다.”

### 2. AI Agent에게 Use Case(Brief), FR/NFR의 초안 요청

“위 프로젝트를 기준으로 Use Case Brief, FR/NFR 초안을 작성해줘.”

### 3. AI Agent에게 기능 범위 분류 요청

“제안한 기능들을 Must, Should, Could, Won't로 나누고, 3주 팀 프로젝트에서 과한 기능은 따로 표시해줘.”

## AI Agent 활용 예시

### 4. AI Agent에게 위험 요소 분석 요청

“이 프로젝트에서 발생할 수 있는 기술적 위험, 일정 위험, 팀 협업 위험, AI Agent 사용 위험을 정리해줘.”

### 5. AI Agent에게 논리적 충돌과 확인 질문 요청

“현재 기획에서 모순되거나 아직 결정되지 않은 부분을 찾아줘. 구현 전에 팀이 답해야 할 질문 목록도 작성해줘.”

## ✓ Elaboration - OOA

### 사람이 AI 결과를 검토하고 범위 확정

AI가 제안한 기능과 위험 요소를 바탕으로

팀이 실제 구현할 **핵심 기능**과 **제외할 기능**을 결정한다.

## ✓ Elaboration - OOA

**“모호한 요구를 구현 가능한 분석 모델로 바꾸고,  
빈틈을 찾는 분석 보조자”**

Inception에서 정리한 요구사항을 더 구체적인 분석 산출물로 발전시킨다.

Use Case 를 정제하고, 설계 초안을 작성한다.

요구사항, Use Case 사이에 연결이 끊긴 부분을 찾아준다.

“무엇을 만들어야 하는가”를 더 명확하게 만드는 분석 보조자로 활용된다.

## ✓ Elaboration - OOA



### AI Agent 활용 예시

#### 1. AI Agent에게 Inception 산출물 제공

“다음은 Inception 단계에서 정한 Use Case Brief, FR/NFR, 위험 목록이다. 이를 기준으로 OOA 산출물을 작성해줘.”

#### 2. AI Agent에게 Use Case를 Casual Format으로 정제 요청

“Use Case Brief를 Casual Format으로 정제해줘. 각 Use Case에 Main Flow, Alternative Flow, Precondition, Postcondition을 포함해줘.”

#### 3. AI Agent에게 System Sequence Diagram과 System Operation 추출 요청

“정제된 Use Case에서 Actor와 System 사이의 이벤트를 추출하고, System Operation 목록을 작성해줘.”

## ✓ Elaboration - OOA



### AI Agent 활용 예시

#### 1. AI Agent에게 Domain Model 초안 요청

“Use Case와 System Operation을 바탕으로 Domain Model에 들어갈 주요 개념, 속성, 관계를 제안해줘.”

#### 2. AI Agent에게 Operation Contract 작성 요청

“각 System Operation에 대해 Operation Contract를 작성해줘. Precondition, Postcondition, 변경되는 도메인 객체를 포함해줘.”

#### 3. AI Agent에게 Traceability Analysis 요청

“FR, Use Case, System Operation, Operation Contract, Test Case 사이의 연결을 표로 만들고, 누락되거나 연결이 약한 부분을 찾아줘.”

## ✓ Elaboration - OOA



### AI Agent 활용 예시

#### 4. AI Agent에게 Domain Model 초안 요청

“Use Case와 System Operation을 바탕으로 Domain Model에 들어갈 주요 개념, 속성, 관계를 제안해줘.”

#### 5. AI Agent에게 Operation Contract 작성 요청

“각 System Operation에 대해 Operation Contract를 작성해줘. Precondition, Postcondition, 변경되는 도메인 객체를 포함해줘.”

#### 6. AI Agent에게 Traceability Analysis 요청

“FR, Use Case, System Operation, Operation Contract, Test Case 사이의 연결을 표로 만들고, 누락되거나 연결이 약한 부분을 찾아줘.”

## ✓ Elaboration - OOA



### 사람이 분석 결과 검토

AI가 만든 Domain Model과 Operation Contract가

실제 게임 규칙과 맞는지 **확인**하고,

잘못된 개념이나 과한 요구사항을 **수정**한다.

## ✓ Elaboration - OOD



### “분석 결과를 구체화하는 설계 보조자”

OOA 산출물을 실제 구현 가능한 설계 산출물로 변환한다.

설계 초안을 만들고, 각 설계 요소가 요구사항과 일관되게 연결되는지 검토한다.

“어떻게 만들 것인가”를 구체화하고, 이후 구현 단계에서 AI Coding Agent가 따라 할 설계 기준을 정리하는 보조자로 활용된다.

## ✓ Elaboration - OOD



### AI Agent 활용 예시

#### 1. AI Agent에게 OOA 산출물과 SRS 제공

“다음 SRS, Use Case, Domain Model, Operation Contract를 기준으로 OOD 산출물을 작성해줘.”

#### 2. AI Agent에게 Use Case를 Fully Dressed Format으로 확장 요청

“Casual Format Use Case를 Fully Dressed Format으로 확장해줘. Stakeholders, Preconditions, Guarantees, Main Success Scenario, Extensions를 포함해줘.”

#### 3. AI Agent에게 UI / Storyboard 초안 요청

“각 Use Case에 필요한 화면을 정리하고, Main Menu, Game Screen, Result Screen의 구성 요소와 화면 전환 흐름을 작성해줘.”

## ✓ Elaboration - OOD



### AI Agent 활용 예시

#### 4. AI Agent에게 System Architecture 대안 요청

“이 프로젝트에 적합한 시스템 아키텍처 대안을 2~3개 제시하고, 각 대안의 장단점과 추천 구조를 설명해줘.”

#### 5. AI Agent에게 Sequence Diagram 작성 요청

“UC-02 Move Character를 기준으로 내부 객체 간 Sequence Diagram을 Mermaid 형식으로 작성해줘.”

#### 6. AI Agent에게 Class Diagram 작성 요청

“Sequence Diagram의 메시지를 바탕으로 필요한 클래스, 속성, 메서드, 관계를 Class Diagram으로 정리해 줘.”

## ✓ Elaboration - OOD



### AI Agent 활용 예시

#### 7. AI Agent에게 Database 또는 Data Schema 작성 요청

“저장/불러오기 기능이 필요한 경우 어떤 데이터를 저장해야 하는지 JSON Schema 또는 Database Schema로 작성해줘.”

#### 8. AI Agent에게 설계 산출물 간 일관성 검사 요청

“Use Case, UI, Sequence Diagram, Class Diagram, Data Schema가 서로 일관되는지 검사하고, 누락된 클래스나 메서드가 있는지 찾아줘.”

## ✓ Elaboration - OOD



### 사람이 최종 설계 선택

AI가 제안한 설계 중

팀이 이해하고 구현할 수 있는 구조를 **선택**하고,

SDS에 **반영**한다.

## ✓ Construction - OOI



**“설계 산출물을 코드와 테스트로 변환하고,  
리뷰와 분석을 통해 구현 품질과 추적성을 유지하는 구현 보조자”**

OOD 산출물을 실제 코드와 테스트로 옮기는 일을 돕는다.

설계에 맞는 코드 초안 작성, 단위 테스트 생성, 시스템 테스트 보완, 코드 리뷰, 정적 분석 결과 해석을 수행할 수 있다.

설계 문서에 없는 기능을 임의로 추가하지 않고, SDS와 테스트 기준을 따라 구현하도록 통제하는 것이 중요하다.

## ✓ Elaboration - OOI



### AI Agent 활용 예시

#### 1. AI Agent에게 SDS와 구현 범위 제공

“다음 SDS를 기준으로 UC-02 Move Character만 구현해줘. 수정 가능한 파일은 game\_controller.py, player.py, collision\_service.py로 제한하고 renderer.py는 수정하지 마.”

#### 2. AI Agent에게 Design to Code Translation 요청

“Class Diagram과 Sequence Diagram을 바탕으로 이번 기능을 구현하기 위해 필요한 클래스, 메서드, 파일 목록을 먼저 정리해줘.”

#### 3. AI Agent에게 코드 작성 요청

“정리한 구현 목록에 따라 코드를 작성해줘. 단, SDS에 없는 기능은 추가하지 말고 기존 구조를 유지해줘.”

## ✓ Elaboration - OOI



### AI Agent 활용 예시

#### 4. AI Agent에게 Unit Test 작성 요청

“moveCharacter 기능에 대한 Unit Test를 작성해줘. 이동 가능 타일, 장애물 타일, 위험 타일의 경우를 모두 포함해줘.”

#### 5. AI Agent에게 System Test 보완 요청

“게임 시작부터 캐릭터 이동, 충돌, 게임 오버 화면까지 이어지는 System Test 시나리오를 작성해줘.”

#### 6. AI Agent에게 Code Review 요청

“작성된 코드가 SDS, Sequence Diagram, Class Diagram을 따르는지 검토해줘. 책임이 잘못 배정된 클래스나 과도하게 긴 함수가 있는지도 찾아줘.”

## ✓ Elaboration - OOI



### AI Agent 활용 예시

#### 7. AI Agent에게 Static Code Analysis 결과 해석 요청

“정적 분석 도구의 결과를 바탕으로 오류, 경고, 코드 품질 문제를 설명하고 수정 우선순위를 제안해줘.”

#### 8. AI Agent에게 구현 Traceability Analysis 요청

“FR, Use Case, Sequence Diagram, Class/Method, 실제 코드, Unit Test, System Test가 연결되어 있는지 표로 정리하고 누락된 부분을 찾아줘.”

## ✓ Elaboration - OOI



### 사람이 코드와 테스트 결과 검토 후 병합

AI가 작성한 코드와 테스트를 사람이 확인하고,

실제 실행 결과와 팀 규칙에 맞는 경우에만 main 브랜치에 병합한다.

## ✓ Transition

**“설계 산출물을 코드와 테스트로 변환하고,  
리뷰와 분석을 통해 구현 품질과 추적성을 유지하는 구현 보조자”**

구현된 시스템을 최종 제출 가능한 상태로 정리하는 일을 돕는다.

최종 검증 및 릴리즈 준비를 수행할 수 있다.

또한 요구사항과 실제 구현 결과가 일치하는지 확인하고, 남은 문제나 제한 사항을 문서화한다.

사용자가 실행하고 평가자가 이해할 수 있는 최종 결과물로 정리하는 마무리 보조자로 활용된다.

## AI Agent 활용 예시

### 1. AI Agent에게 최종 구현 결과와 제출/배포 조건 제공

“현재 시스템 구현이 완료되었고, 최종 제출 또는 배포를 준비해야 한다. 실행 환경, 제출물, 사용자 안내 문서, 테스트 결과를 기준으로 Transition 단계에서 점검해야 할 항목을 정리해줘.”

### 2. AI Agent에게 최종 테스트 시나리오 작성 요청

“사용자가 시스템을 처음 실행한 뒤 주요 기능을 사용하는 전체 흐름을 기준으로 최종 System Test 시나리오를 작성해줘.”

### 3. AI Agent에게 결함 및 개선 항목 우선순위 정리 요청

“다음 버그와 개선 항목을 심각도, 사용자 영향도, 수정 난이도 기준으로 분류하고, 릴리즈 전에 반드시 해결해야 할 항목을 구분해줘.”

## AI Agent 활용 예시

### 4. AI Agent에게 사용자 매뉴얼 작성 요청

“시스템 실행 방법, 주요 기능 사용법, 입력 방법, 제한 사항, 오류 발생 시 대처 방법을 포함한 사용자 매뉴얼을 작성해줘.”

### 5. AI Agent에게 릴리즈 노트 작성 요청

“이번 버전의 릴리즈 노트를 작성해줘. 포함된 기능, 수정된 문제, 알려진 제한 사항, 실행 환경을 정리해줘.”

### 6. AI Agent에게 배포 절차 점검 요청

“최종 배포 또는 제출 전에 필요한 절차를 체크리스트로 정리해줘. 실행 파일, 의존성, 설정 파일, 문서, 테스트 결과가 누락되지 않았는지 확인해줘.”

## ✓ Transition



### AI Agent 활용 예시

#### 7. AI Agent에게 발표 또는 인수 자료 보완 요청

“평가자나 사용자가 시스템의 목적, 주요 기능, 사용 방법, 구현 결과를 쉽게 이해할 수 있도록 발표 자료 또는 인수 자료의 흐름을 점검해줘.”

#### 8. AI Agent에게 최종 Traceability Analysis 요청

“요구사항, 설계 문서, 구현 코드, 테스트 결과가 서로 연결되어 있는지 최종 점검해줘. 요구사항은 있지만 구현 되지 않은 부분이나, 구현은 되었지만 문서에 없는 부분을 찾아줘.”

## ✓ Transition

### **사람이 최종 실행과 제출 여부 결정**

AI가 만든 체크리스트와 문서를 바탕으로

사람이 직접 시스템을 실행하고,

테스트 결과와 제출물을 확인한 뒤

최종 배포 또는 제출을 결정한다.

## ✓ Operation

### “운영 중 발생하는 문제와 변경 요구를 분석하고, 장애 대응·유지보수·개선 작업을 체계화하는 운영 보조자”

배포 이후 시스템이 안정적으로 사용될 수 있도록 운영과 유지보수 업무를 돕는다.

로그 분석, 장애 원인 추정, 버그·피드백 정리, 회귀 테스트 추천을 통해 운영 문제를 체계화한다. 또한 변경 영향 범위를 추적해 무분별한 수정으로 인한 품질 저하를 줄인다.

운영 중 발생하는 정보를 분석하고 유지보수 작업을 안전하게 진행하도록 돕는 운영 보조자로 활용된다.

## AI Agent 활용 예시

### 1. AI Agent에게 운영 상황과 문제 정보 제공

“현재 시스템이 배포 또는 제출 이후 사용 중이며, 일부 사용자에게 오류가 발생했다. 다음 오류 메시지, 실행 환경, 발생 상황을 바탕으로 문제 원인을 분석해줘.”

### 2. AI Agent에게 로그와 오류 메시지 분석 요청

“다음 로그에서 오류 발생 지점, 가능한 원인, 재현 조건을 추정해줘. 단정하지 말고 가능성이 높은 원인부터 정리해줘.”

### 3. AI Agent에게 버그 리포트 정리 요청

“사용자 제보와 로그를 바탕으로 버그 리포트를 작성해줘. 발생 조건, 예상 동작, 실제 동작, 심각도, 재현 절차를 포함해줘.”

## AI Agent 활용 예시

### 4. AI Agent에게 장애 대응 우선순위 정리 요청

“다음 문제들을 사용자 영향도, 심각도, 수정 난이도 기준으로 분류하고, 먼저 해결해야 할 항목을 제안해줘.”

### 5. AI Agent에게 수정 영향 범위 분석 요청

“이 버그를 수정하려면 어떤 요구사항, Use Case, 클래스, 테스트가 영향을 받을 수 있는지 분석해줘.”

### 6. AI Agent에게 유지보수 작업 단위 분해 요청

“이 문제를 해결하기 위한 작업을 작은 단위로 나눠줘. 각 작업마다 수정 대상, 확인할 테스트, 주의할 점을 정리해 줘.”

## ✓ Transition



### AI Agent 활용 예시

#### 7. AI Agent에게 회귀 테스트 항목 추천 요청

“이번 수정으로 기존 기능이 깨질 수 있는 부분을 찾아주고, 반드시 다시 실행해야 할 회귀 테스트 목록을 작성해줘.”

#### 8. AI Agent에게 사용자 피드백 분류 요청

“다음 사용자 피드백을 버그, 개선 요청, 사용성 문제, 문서 보완 요청으로 분류하고 우선순위를 제안해줘.”

#### 9. AI Agent에게 운영 문서 보완 요청

“이번 문제와 해결 방법을 바탕으로 FAQ, Troubleshooting Guide, 운영 기록 문서를 업데이트해줘.”

## ✓ Operation

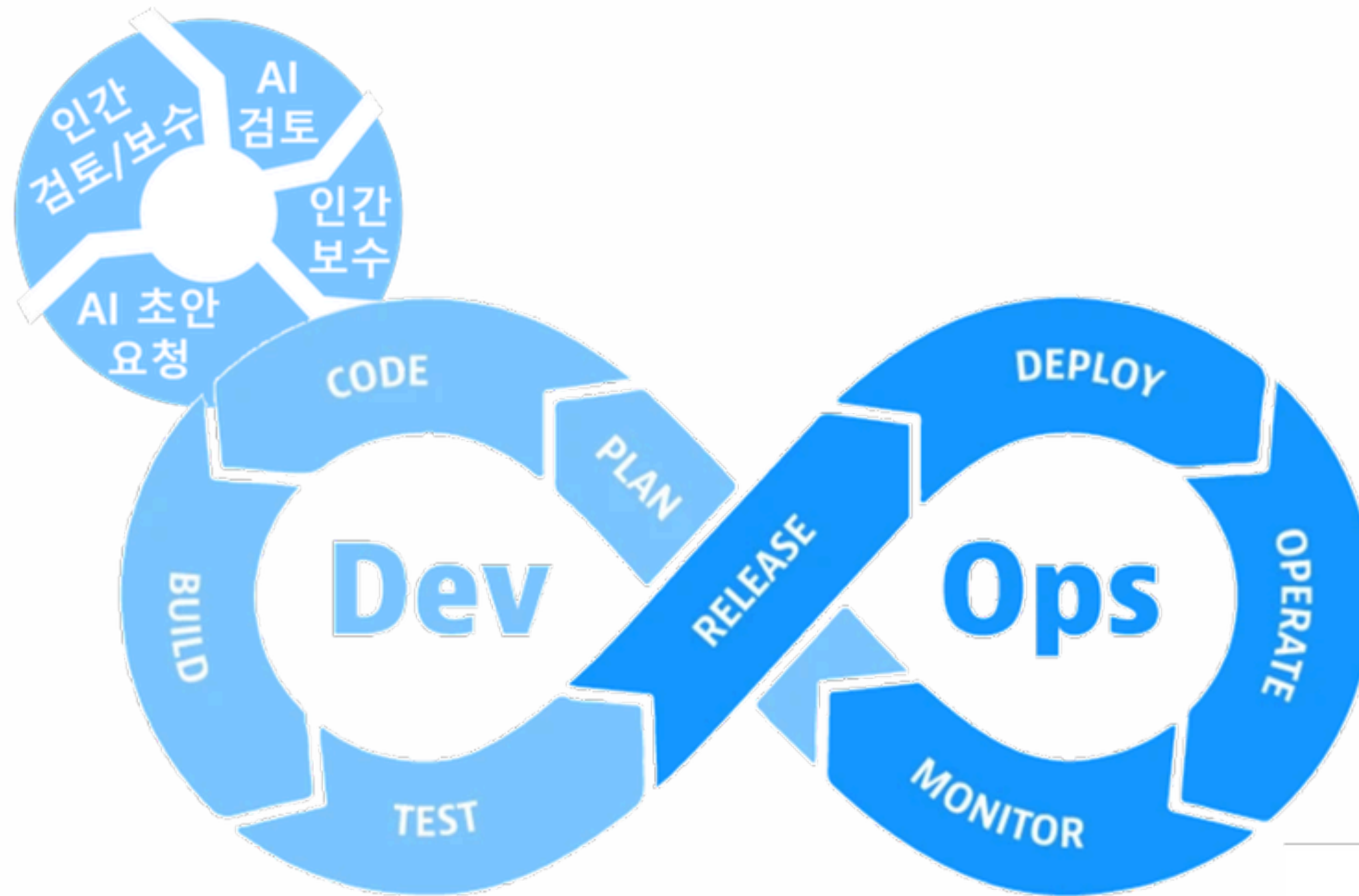


### 사람이 최종 수정 및 운영 반영 여부 결정

AI Agent가 분석한 원인, 영향 범위, 테스트 결과를 바탕으로

사람이 실제 수정 여부와 배포 여부를 결정한다.

# ✓ 큰 흐름



## ✓ 제안



AI가 생성한 코드를 검증할 수 있는 플랫폼/도구가 필요

Ex) 설계문서대로 제작했는지 확인하기 위한 도구 -  
호출된 함수/객체 + Class Diagram or SD

AI가 생성한 코드를 기반으로 AI가 제작한 “사후 클래스 다이어그램”을 이용해,  
기존에 제작한 클래스 다이어그램과 비교하며 코드를 분석하는 방식 가능

AI가 프로그램의 전체적인 내용을 파악할 수 있는 Context 파일 양식 필요