# OOAD:
# #7 Optimized and Balanced SW Development Process

202211318 엄정석 / 202211334 이동훈 / 202211384 차민우 / 202211392 최환

# Contents

# About AI

## 기존 방법론의 한계점

# Limitations

## 1. 문서화에 자원이 과도하게 소모된다.

→ 산출물의 내용을 구상하는 작업과 별개로,
그 내용을 정해진 양식에 맞춰 정리하는 데 추가 비용이 발생한다.

## 2. 변경사항 반영이 어렵다.

→ 방법론을 진행하는 도중 수정사항이 생기면 그와 연관된 이전 문서들을 모두 갱신해야 한다.
진행한 단계가 많을수록 수정 대상 문서가 늘어나 시간 소모가 커진다.

# About AI

## AI의 장단점

# Pros

- **속도가 빠르다.**

  주어진 프롬프트에 대해 사람보다 훨씬 빠르게 작업을 수행한다.

- **프로젝트의 컨텍스트를 유지할 수 있다.**

  동일 세션 내에서/별도의 memory,md 파일을 구축함으로써

  작업 맥락을 유지할 수 있다.

# Cons

- **정확하지 않다.**

  작업을 완료했다고 응답하지만 할루시네이션이 발생해

  실제 결과물이 비어있거나 빈약한 경우가 많다.

- **중간점검이 필요하다.**

  진행 과정에서 요청하지 않은 부분들을 멋대로 생성하거나

  요구사항을 임의로 변경하기도 한다.

# About AI

## AI 적용 방안

# Apply

1. 자유로운 정보 입력을 통한 문서 표준화 자동화

2. 연관 문서 추적 및 동기화 기반의 일관성 유지와 사람의 검수

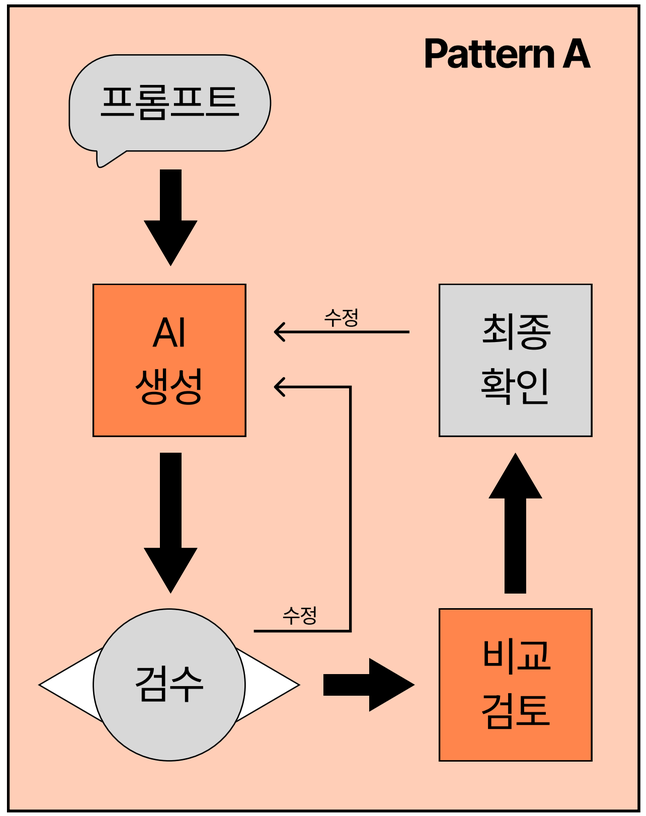
3. 과거 산출물 기반의 지능형 품질 검증

⇒ 사람은 “**의도**”와 “**설계**”를 담당하고,
AI는 **문서화**, **수정**, **변환**, **구현**을 담당한다.

# About AI

## 공통 패턴

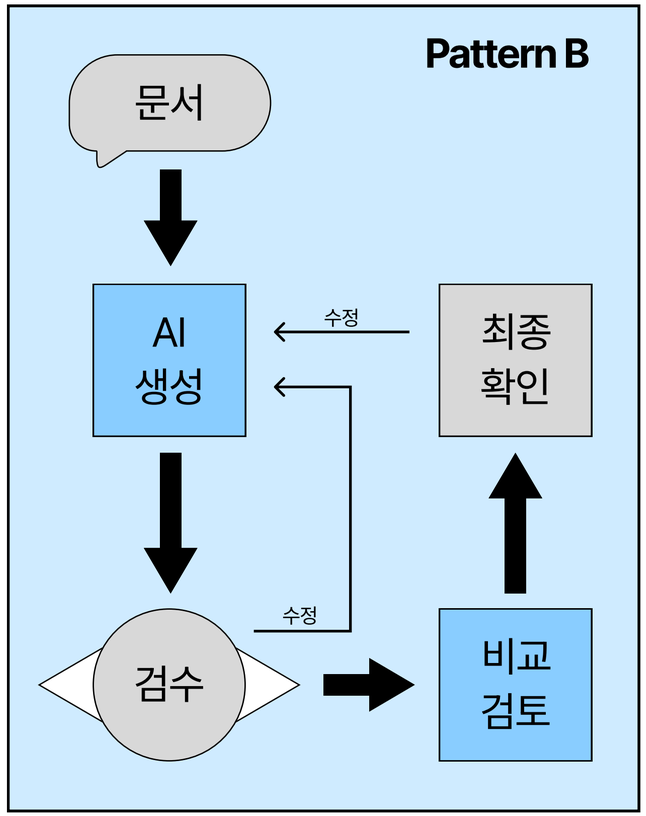
# 패턴 A




## 사람이 의도를 작성하고 AI가 정형화

1. 사람이 양식에 구애받지 않고 캐주얼한 문체로 작성한다.
2. AI가 정식 문서 형태로 리포맷팅하고 버전을 명시·관리한다.
3. 산출물을 사람이 검수한다.
4. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

# 패턴 B




## AI가 AI-Understandable 문서를 기반으로 생성

1. 사람이 작성한 AI-Understandable 문서를 입력으로 AI가 산출물을 생성한다.
2. 산출물을 사람이 검수한다.
3. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
4. 사람이 최종 확인하고 필요 시 수정한다.

# Inception

## AI-Understandable Project Overview 문서

# AI-Understandable Project Overview 문서

- 해당 프로젝트에서 어떤 문서들을 채택하여 사용하는지 명시한다.
- 채택한 각각의 문서에 대한 양식을 명시한다.
- 프로젝트에서 사용할 기술 스택을 정리한다.

**⇒ AI로 하여금 프로젝트 전반의 내용을 파악할 수 있도록 한다.**

# Inception

## Glossary

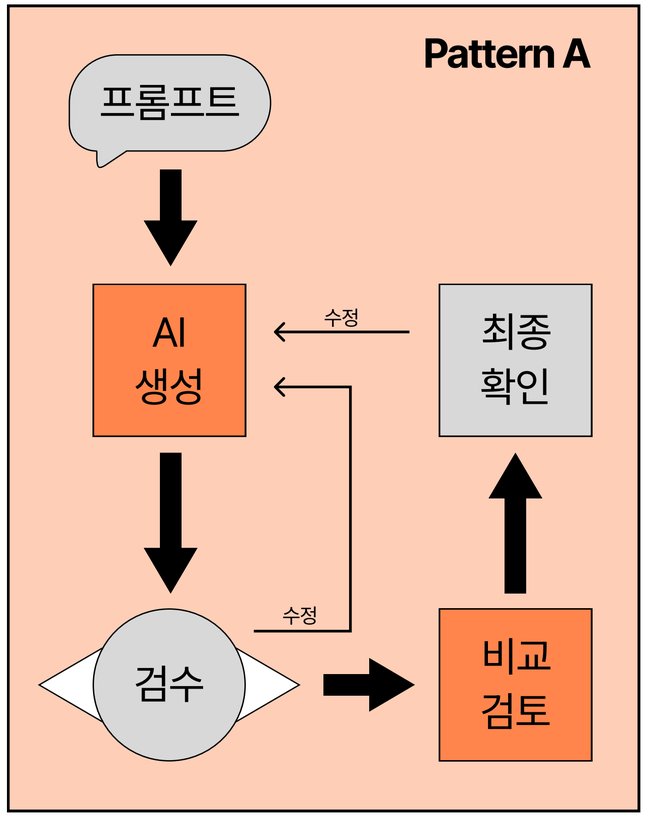
# Glossary

- 프로젝트에서 사용하는 핵심 용어와 그 정의를 정리한다.
- 동일 개념에 대한 표기를 하나로 고정해,
  AI가 용어를 혼동하거나 임의로 바꾸는 할루시네이션을 억제한다.

# OOAD

FR/NFR, Use-Case

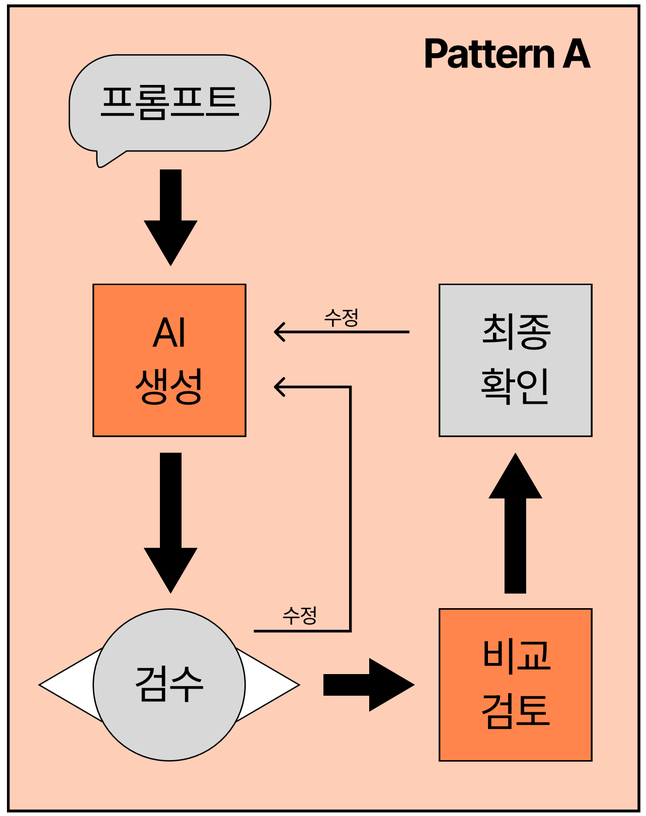
# FR/NFR




## 패턴 A를 따라 구현한다.

1. 양식에 구애받지 않고 **FR/NFR에 대해 캐주얼한 문체**로 작성한다.
2. AI가 정식 문서 형태로 리포맷팅하고 버전을 명시·관리한다.
3. AI가 리포맷팅한 FR/NFR을 사람이 검수한다.
4. AI가 관련 산출물(*AI-Understandable Project Overview, Glossary*)과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

# Use-Case



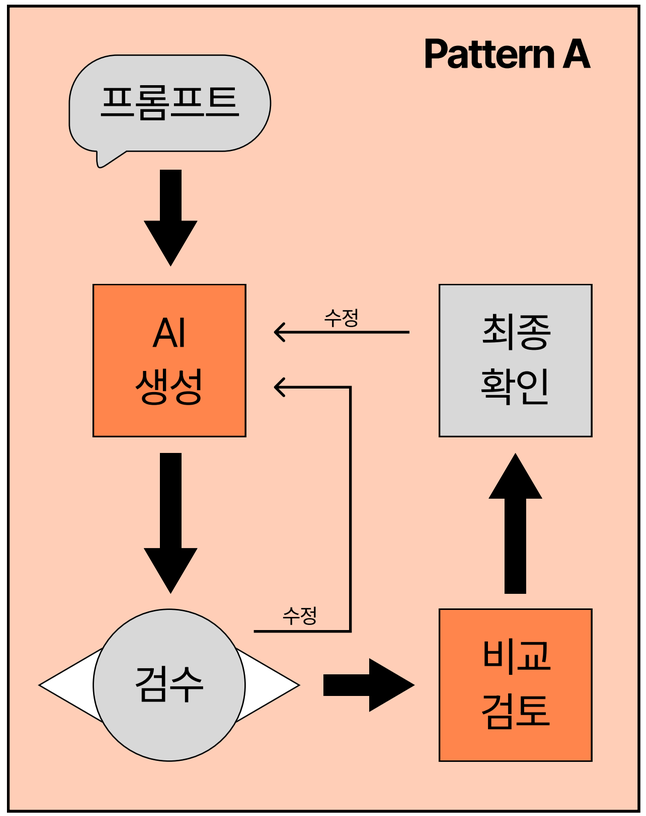

## 패턴 A를 따라 구현한다.

1. 양식에 구애받지 않고 **Use-Case에 대해 캐주얼한 문체**로 작성한다.
2. AI가 정식 문서 형태로 리포맷팅하고 버전을 명시·관리한다.
3. AI가 리포맷팅한 Use-Case를 사람이 검수한다.
4. AI가 관련 산출물(*AI-Understandable Project Overview, Glossary, FR/NFR*)과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

# OOAD

## System Sequence Diagram, Domain Model

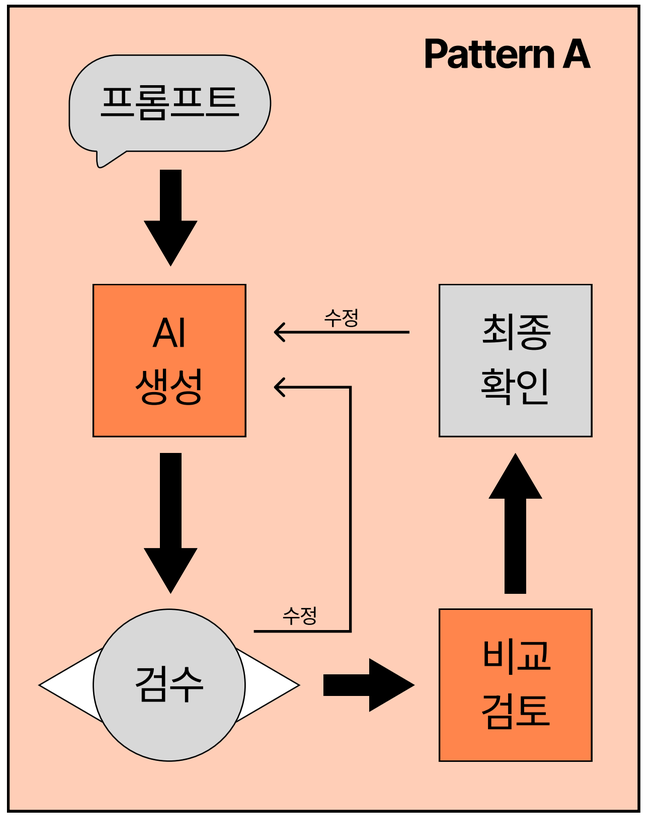
# System Sequence Diagram




## 패턴 A를 따라 구현한다.

1. 양식에 구애받지 않고 **S.S.D.에 대해 캐주얼한 문체**로 작성한다.
2. AI가 다이어그램을 생성하고 버전을 명시·관리한다.
   다이어그램은 AI모델과 사용자 모두가 수정할 수 있는
   텍스트 기반 다이어그램 툴(PlantUML)을 사용한다.
3. AI가 산출한 S.S.D.을 사람이 검수한다.
4. AI가 관련 산출물(*AI-Understandable Project Overview, Glossary, FR/NFR, Use-Case*)과
   비교해 누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

# Domain Model




## 패턴 A를 따라 구현한다.

1. 양식에 구애받지 않고 **Domain Model에 대해 캐주얼한 문체**로 작성한다.
2. AI가 다이어그램을 생성하고 버전을 명시·관리한다.
   도메인 모델은 AI모델과 사용자 모두가 수정할 수 있는
   텍스트 기반 다이어그램 툴(PlantUML)을 사용한다.
3. AI가 산출한 Domain Model을 사람이 검수한다.
4. AI가 관련 산출물(*AI-Understandable Project Overview, Glossary, FR/NFR, Use-Case, SSD*)과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

# OOAD

## AI-Understandable
## Class Component Overview

# AI-Understandable Class Component Overview

- Class Diagram에 들어가는 Attribute, Operation을 나열만 해서는

  인공지능이 어떤 역할을 하고 왜 존재하는지 이해할 수 없고 억측을 할 가능성이 존재한다.

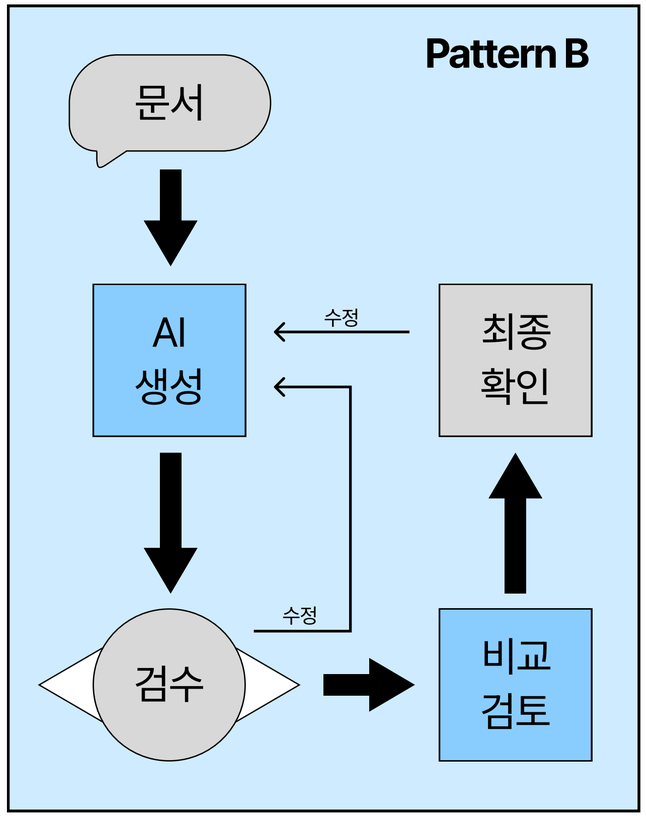
- 이를 해결하기 위해 ‘**각각의 Attribute와 Operation이 왜 존재하는지**’,

  ‘**어떤 역할과 기능을 가지는지**’를 작성하는 문서이다.

# OOAD

## Sequence Diagram, Class Diagram

# Sequence Diagram




## 패턴 B를 따라 구현한다.

1. **AI-Understandable Class Component Overview를 기반**으로

   **주요 시나리오의 상호작용을 캐주얼한 문체**로 작성한다.

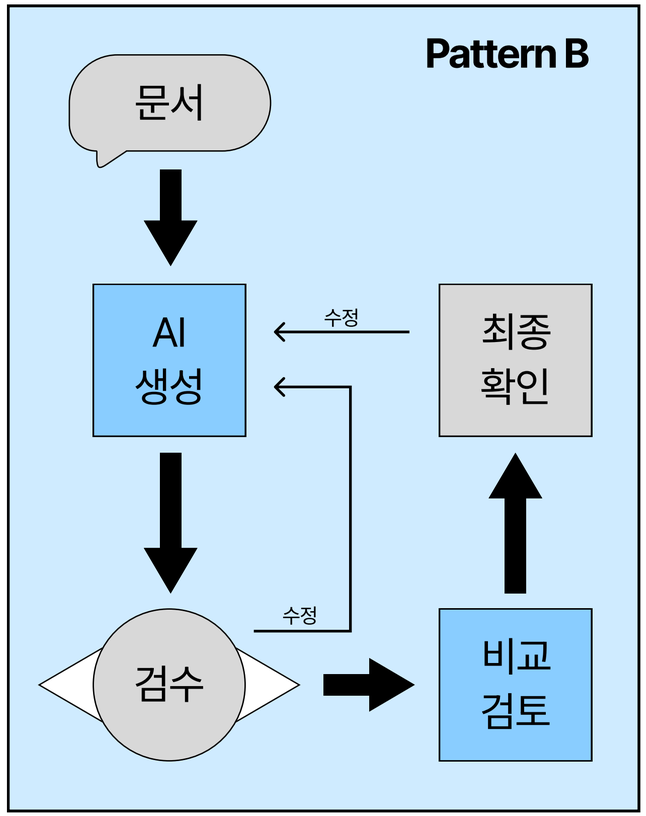
2. AI가 이를 Sequence Diagram으로 정형화하고 버전을 명시·관리한다.
   다이어그램은 AI모델과 사용자 모두가 수정할 수 있는
   텍스트 기반 다이어그램 툴(PlantUML)을 사용한다.

3. AI가 산출한 Sequence Diagram을 사람이 검수한다.

4. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
   *산출물: *AI-Understandable Project Overview*, *Glossary*, *OOA* 산출물,
   *AI-Understandable Class Component Overview*

5. 사람이 최종 확인하고 필요 시 수정한다.

# Class Diagram




## 패턴 B를 따라 구현한다.

1. **AI-Understandable Class Component Overview를 기반**으로

   **주요 시나리오의 상호작용을 캐주얼한 문체**로 작성한다.

2. AI가 산출했던 Sequence Diagram에서 사용한 Operation을

   Class Diagram에 반영하여 생성하고 버전을 명시·관리한다.
   다이어그램은 AI모델과 사용자 모두가 수정할 수 있는
   텍스트 기반 다이어그램 툴(PlantUML)을 사용한다.

3. AI가 산출한 Class Diagram을 사람이 검수한다.

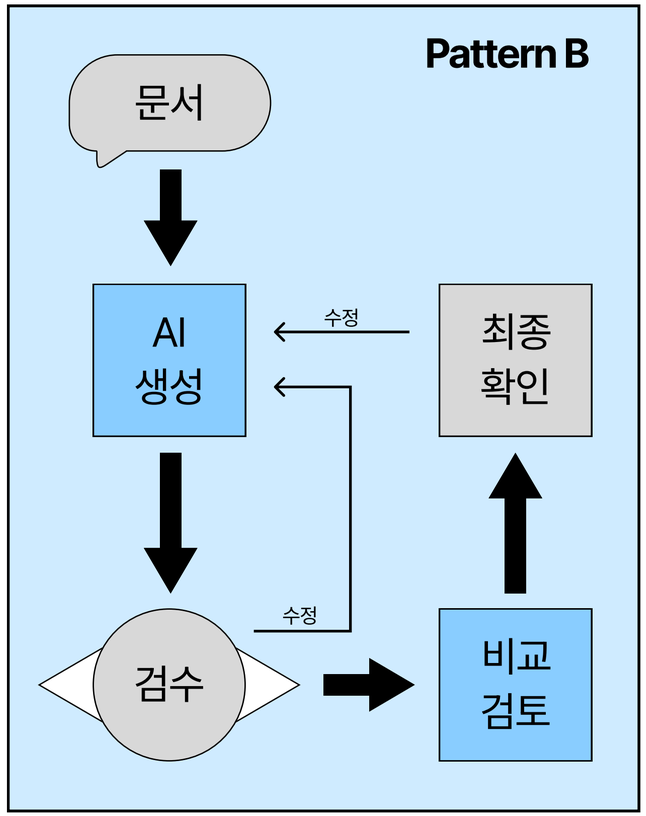
4. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
   *산출물: *AI-Understandable Project Overview*, *Glossary*, *OOA* 산출물,
   *AI-Understandable Class Component Overview*, *Sequence Diagram*

5. 사람이 최종 확인하고 필요 시 수정한다.

# OOI

## Unit Test

# Unit Test




## 패턴 B를 따라 구현한다.

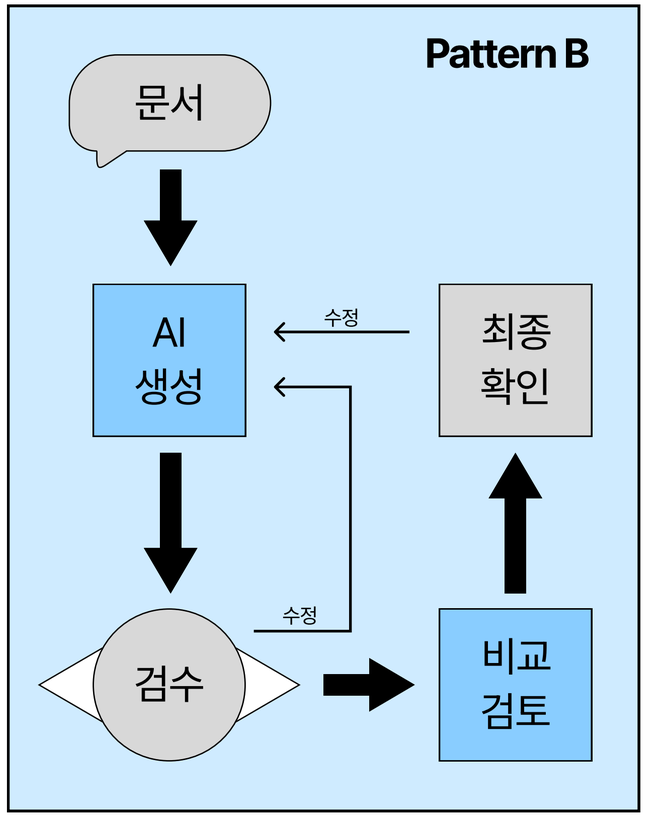
1. Unit Test가 무엇을 테스트하고 어떤 방식을 사용하는지 정리한

   **Unit Test 문서**(*AI-Understandable* 문서)를 작성한다.
2. AI가 Unit Test 문서를 기반으로 Unit Test를 생성한다.
3. 산출물을 사람이 검수한다.
4. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.
   *산출물: *AI-Understandable Project Overview, Glossary, OOAD* 산출물
5. 사람이 최종 확인하고 필요 시 수정한다.

# OOI

Coding

# Coding



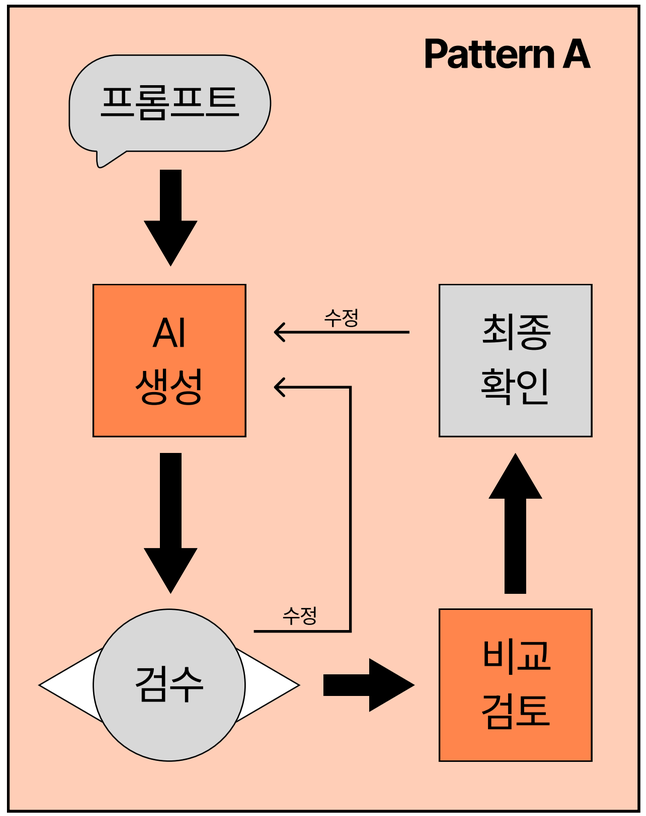

## 패턴 B를 따라 구현한다.

1. 사람이 **특정 기능의 코드 구현을 요청**한다.
2. AI가 **이전에 산출한 문서들을 기반**으로 코드를 생성한다.
3. 생성한 코드를 사람이 검수한다.
4. 위 과정을 **전체 시스템을 완성할 때까지 반복**한다.
5. AI가 코드를 생성할 때마다 Code Report를 작성하도록 하여,
   각 객체·함수를 왜, 어떻게 구현했는지 기술하게 하여
   코드 검수에 걸리는 시간을 단축한다.

# OOI

## System Test

# System Test




## 패턴 A를 따라 구현한다.

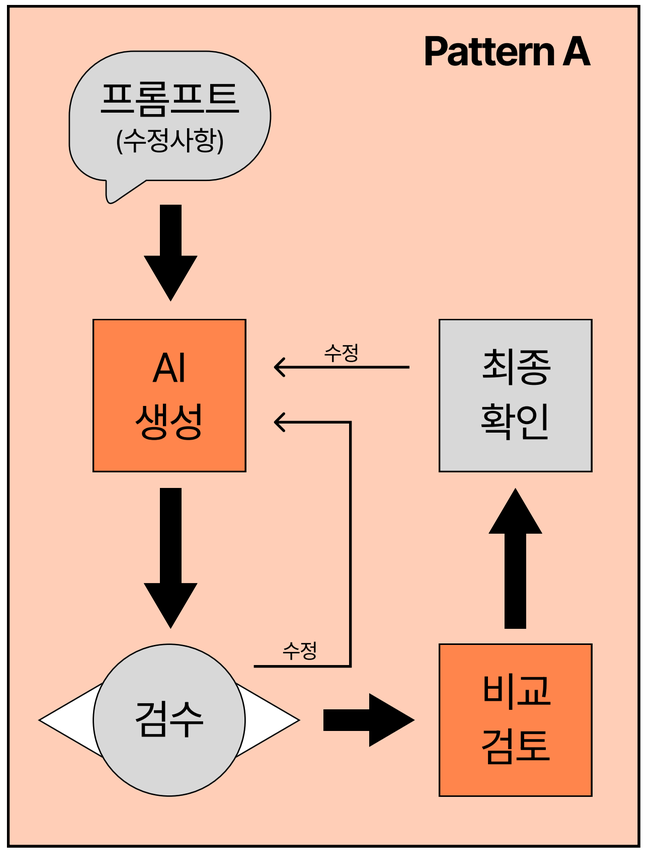
1. 양식에 구애받지 않고 **System Test에 대해 캐주얼한 문체**로 작성한다.
2. AI가 프롬프트를 기반으로 System Test를 생성하고 버전을 명시·관리한다.
3. AI가 산출한 System Test를 사람이 검수한다.
4. AI가 관련 산출물과 비교해 누락·오류를 검토하여 수정 사항을 정리한다.

   *산출물: *AI-Understandable Project Overview*, *Glossary*, *OOAD* 산출물, *Unit Test*, *Code**
5. 사람이 최종 확인하고 필요 시 수정한다.

# OOI

## Deployment

# Deployment




## 패턴 A를 따라 구현한다.

1. 수정해야 할 부분에 대해 캐주얼한 문체로 작성한다.
2. AI가 수정 사항을 기반으로 수정하고 버전을 명시·관리한다.
3. AI가 산출한 Release Note(변경 이력)를 사람이 검수한다.
4. AI가 이전 버전과 수정한 내용을 참고해
   누락·오류를 검토하여 수정 사항을 정리한다.
5. 사람이 최종 확인하고 필요 시 수정한다.

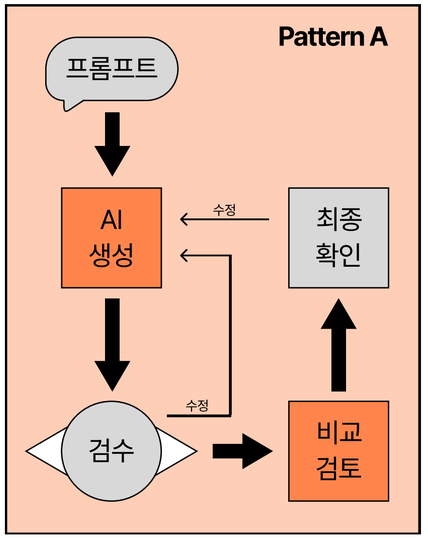
# Diagram

## Inception

- AI-Understadable Project Overview 문서
- Glossary

## OOA

- FR/NFR
- Use-Case
- System Sequence Diagram
- Domain Model

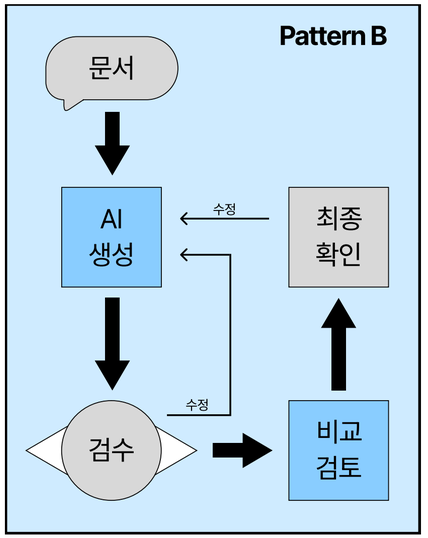



## OOD

- AI-Understandable Class Component Overview
- Sequence Diagram
- Class Diagram

## OOI

- Unit Test
- Coding
- System Test
- Deployment (Release Note)



# 감사합니다.

# Diagram 원본(수정용)