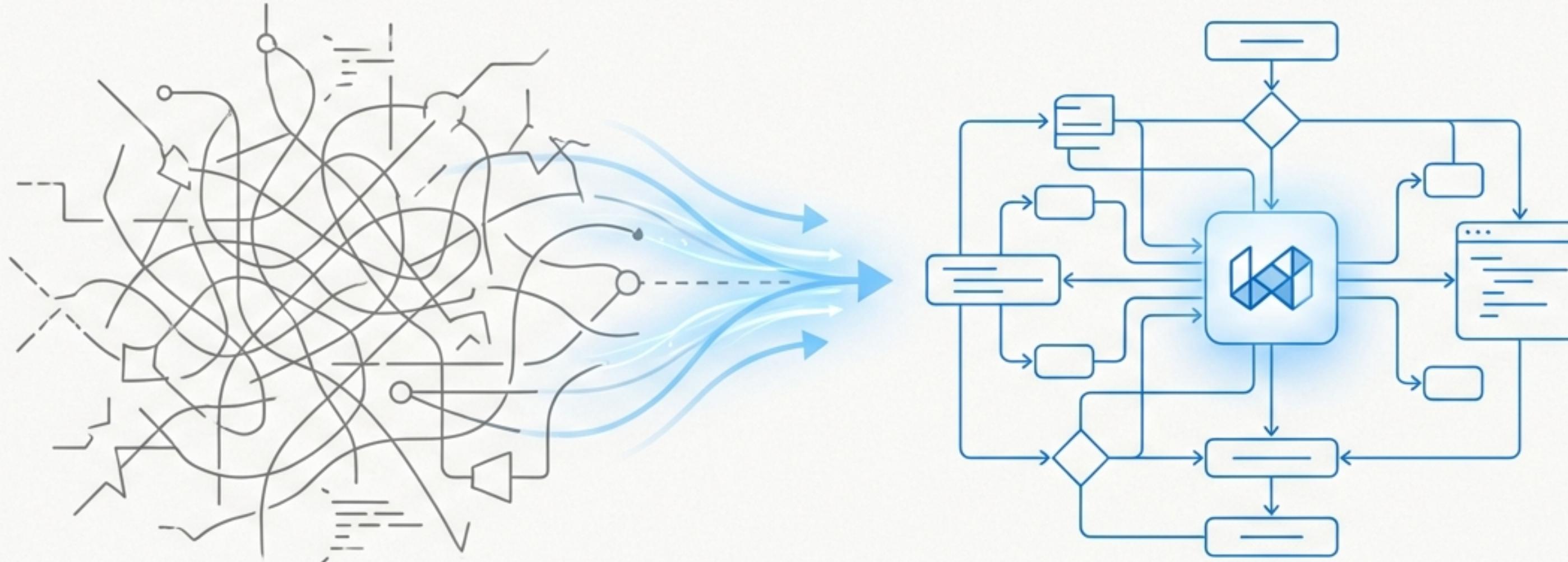


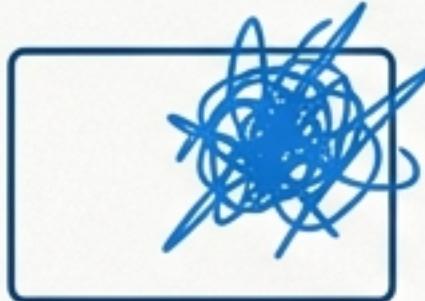
# 원더 무브 연구소

# Claude Code 업무 동화 플러그인



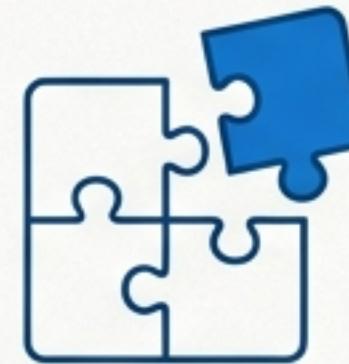
어떤 상황에서도 동일한 개발 품질을 보장하는 AI 개발 도구

# AI 코딩의 보이지 않는 비용: 예측 불가능한 품질



## 새 프로젝트

설계 없이 즉흥적으로 코딩하여  
아키텍처가 무시됩니다.



## 기존 프로젝트

기존 패턴을 무시하여  
코드 스타일이 충돌합니다.



## 작업 재개

이전 맥락을 잊어버려  
핵심 규칙을 위반합니다.



← 기술 부채 (Technical Debt)

이 모든 것이 일관성 없는 코드와 누적되는 기술 부채로 이어집니다.

# 해결책: 일관성을 '강제'하는 지능형 개발 시스템

## 워크플로우 강제

아이디어부터 구현까지 체계적인  
프로세스를 따릅니다.



## 컨텍스트 자동 학습

기존 프로젝트의 스타일과 규칙을  
스스로 파악합니다.

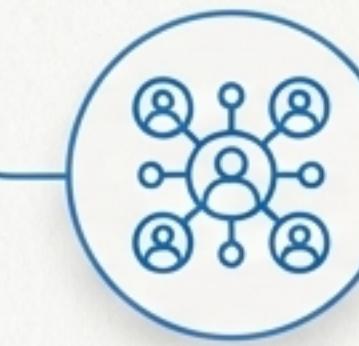


## 품질 규칙 자동 적용

개발자가 신경 쓰지 않아도  
품질 기준을 준수합니다.



## Claude Code Plugin

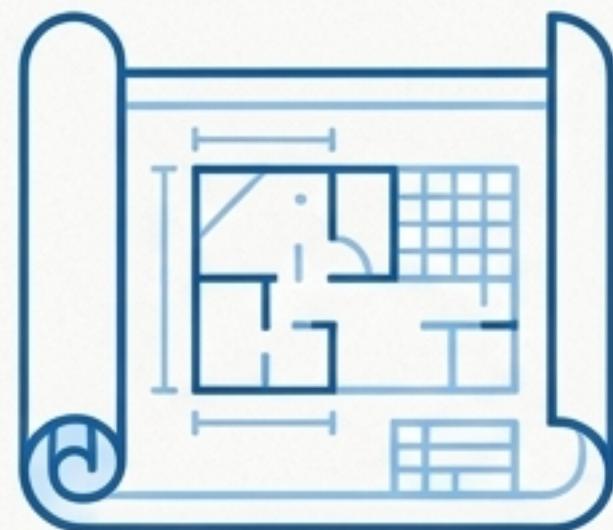


## 진행 상황 투명화

모든 작업이 실시간으로  
추적되고 공유됩니다.

Claude Code 플러그인은 AI 개발에 체계와 규칙을 부여합니다.

# 언제, 누가, 어떻게 시작하든 동일한 결과물



/dev plan

## 새 프로젝트

체계적인 기획과 설계를 통해 첫 단추를 올바르게 끼웁니다.



/onboard

## 기존 프로젝트 합류

단 몇 분 만에 프로젝트 구조와 패턴을 학습하여 즉시 기여합니다.



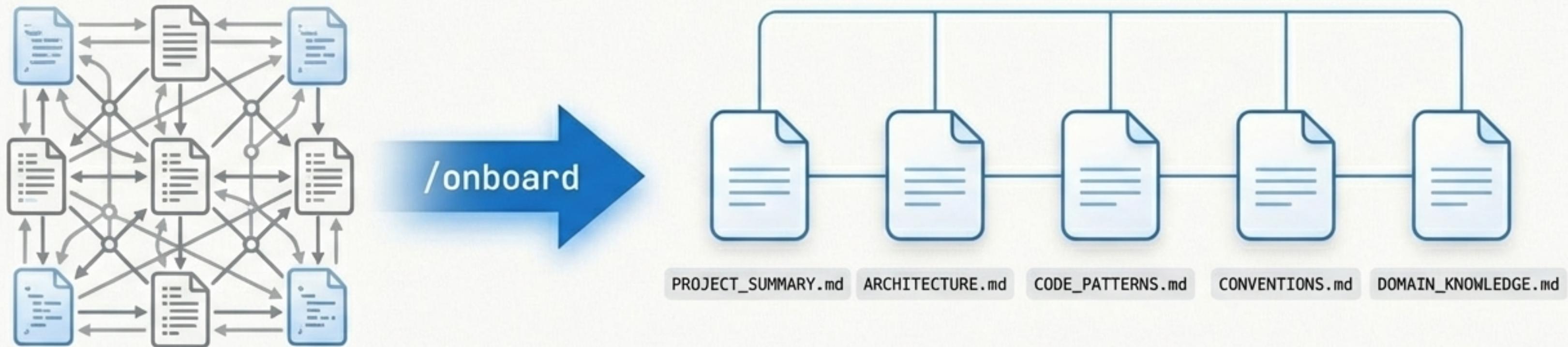
/restore-context

## 중단된 작업 재개

중단 지점의 모든 맥락을 즉시 복원하여 재작업을 방지합니다.

# 10분 만에 프로젝트 전문가 되기: `/onboard`

명령어 하나로 코드베이스를 분석하여 5개의 핵심 컨텍스트 문서를 자동 생성합니다.



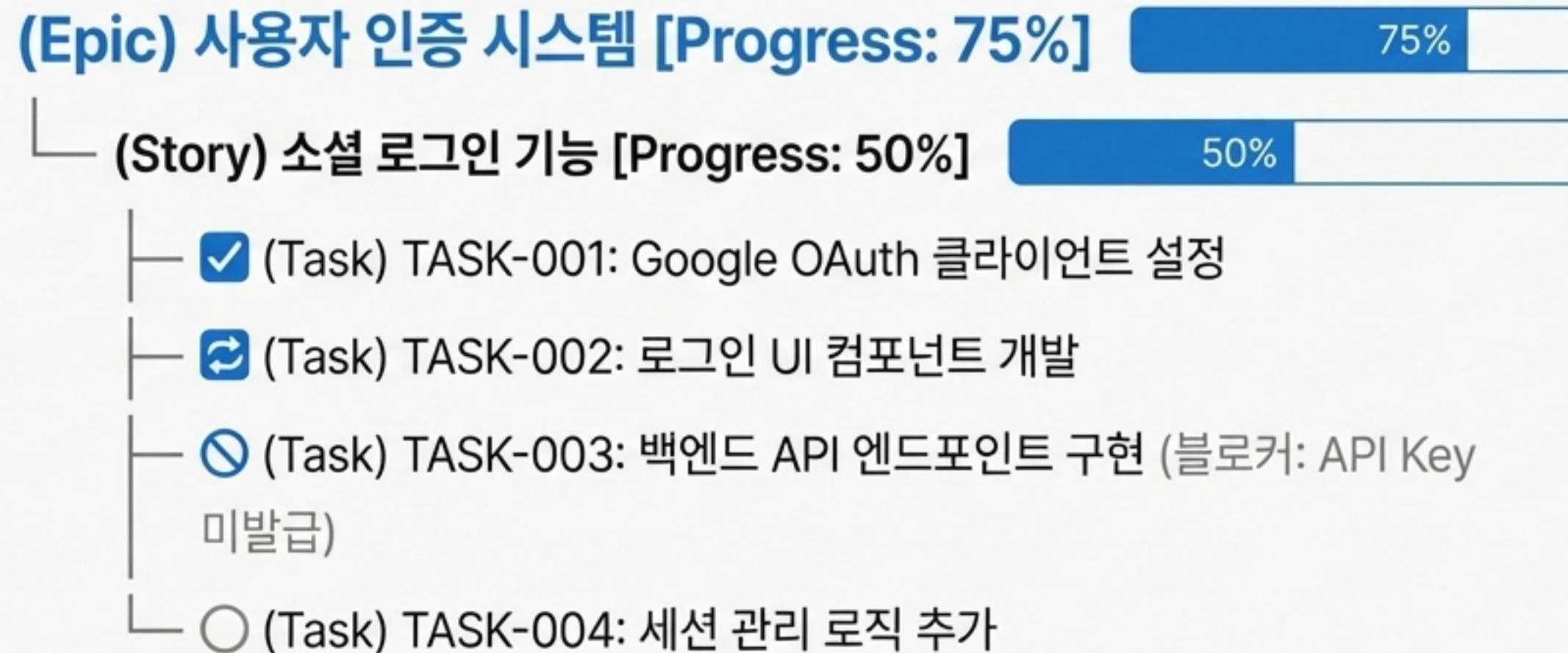
결과: AI가 프로젝트의 'DNA'를 학습하여, 이후 모든 코드 생성 시  
기존 스타일과 완벽한 일관성을 유지합니다.

# 아이디어를 코드로 변환하는 체계적인 워크플로우



# 모든 진행 상황의 시각화: Worktree

개발 진행 상황을 실시간으로 추적하고 시각화하는 CLI 기반 작업 트리입니다.



- **구조**
  - `Epic → Story → Task`의 계층 구조로 작업을 관리합니다.
- **상태**
  - ✓ 완료 | ⟳ 진행중 | 🚫 블로커 | ○ 대기
- **자동화**
  - 진행률이 자동으로 계산되고, 명령어(`start`, `done`, `block`)로 상태를 쉽게 변경합니다.

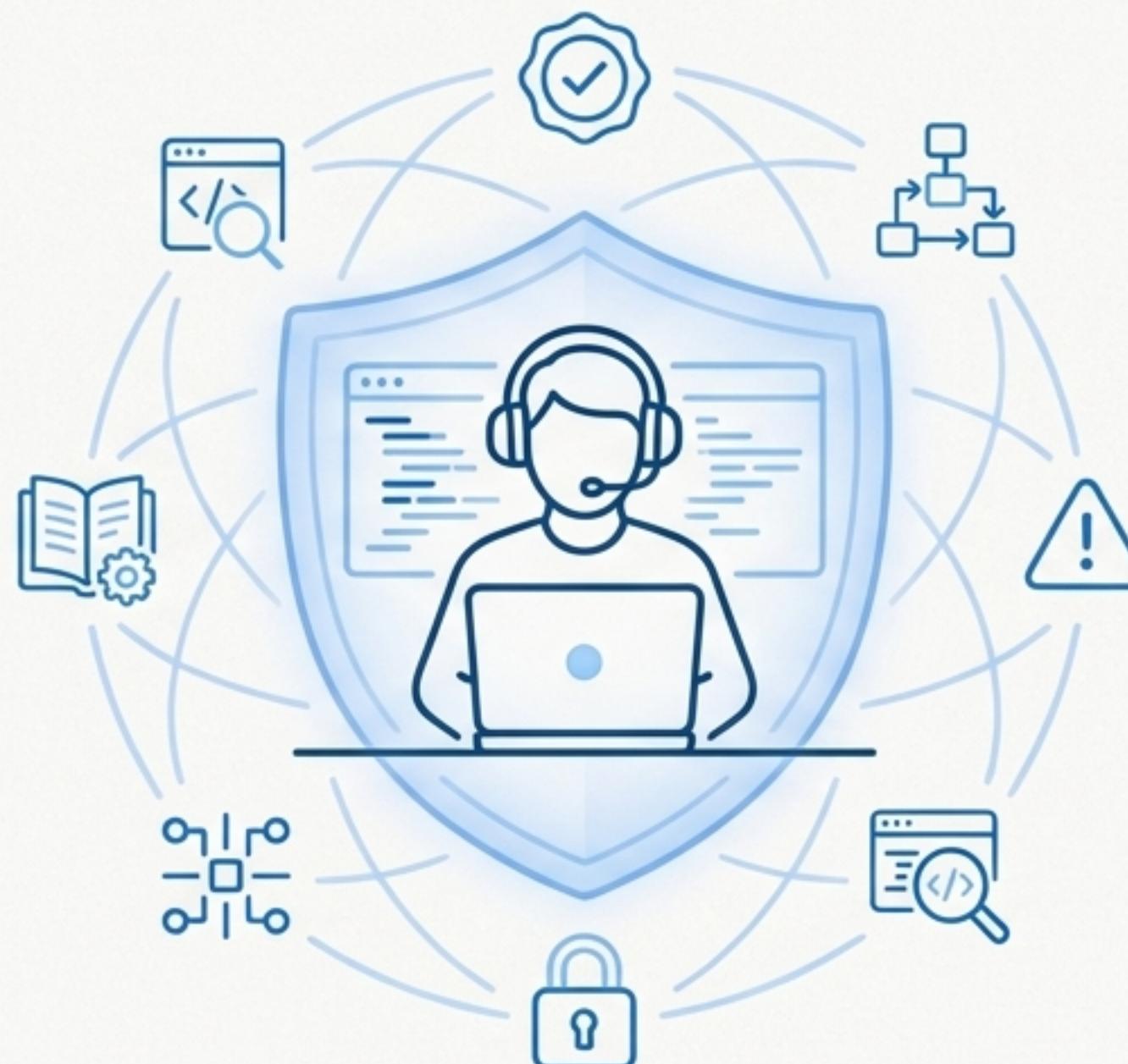
# 개발과 관리를 잇는 실시간 JIRA 양방향 동기화



**가치:** PM과 관리자는 JIRA 대시보드에서 코드를 보지 않고도 팀의 실제 진행 상황을 실시간으로 파악할 수 있습니다.

# 신경 쓰지 않아도 지켜지는 품질: 자동 활성화 스킬

개발자의 특정 키워드나 행동을 감지하여 9개의 내장 스킬이 자동으로 품질 규칙을 적용합니다.



## `code-quality`

코드 생성 시 '300줄 제한' 및 '주석 필수' 규칙을 강제합니다.



## `clean-architecture`

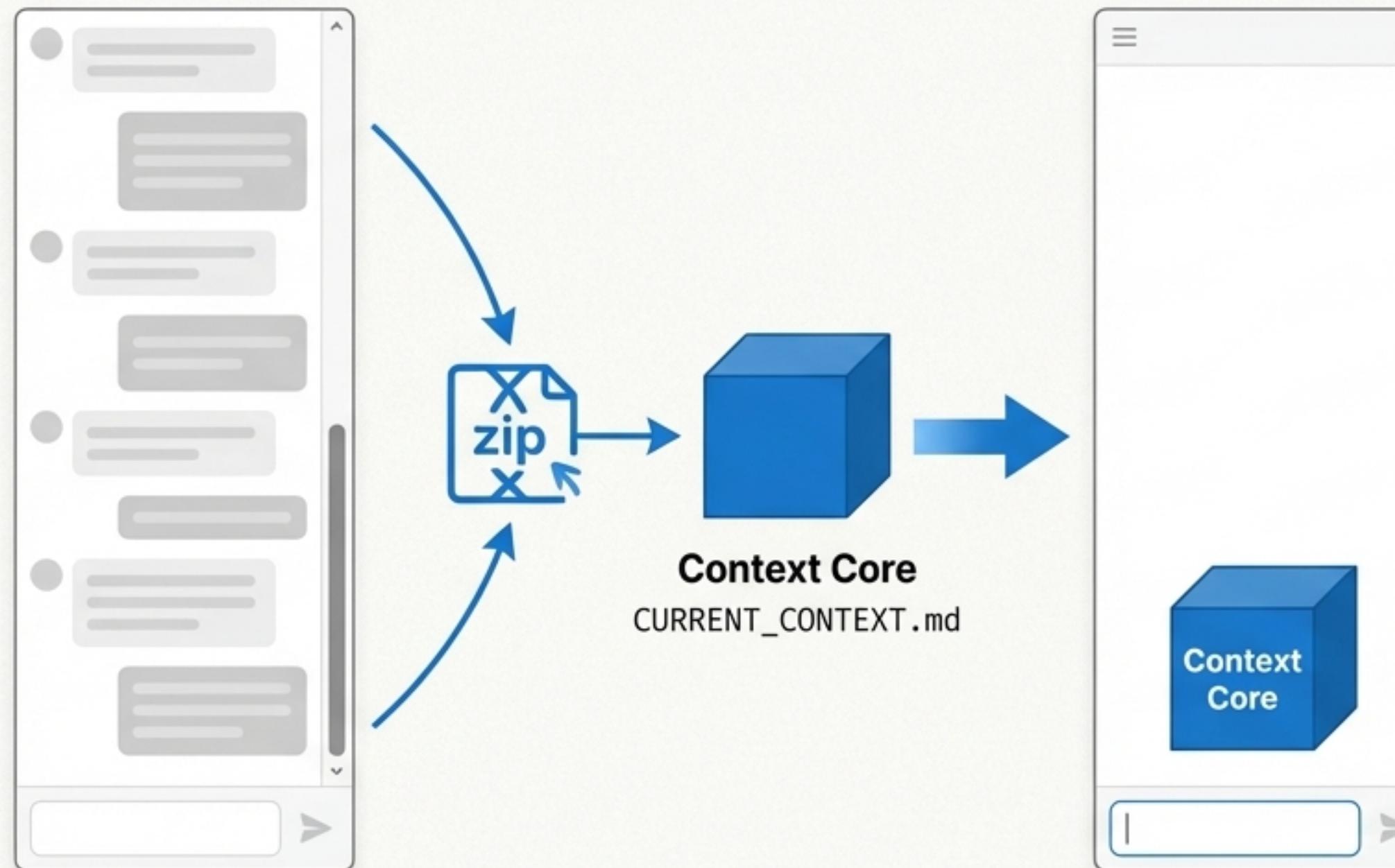
레이어 간 의존성 규칙 위반 시 즉시 경고합니다.



## `best-practices`

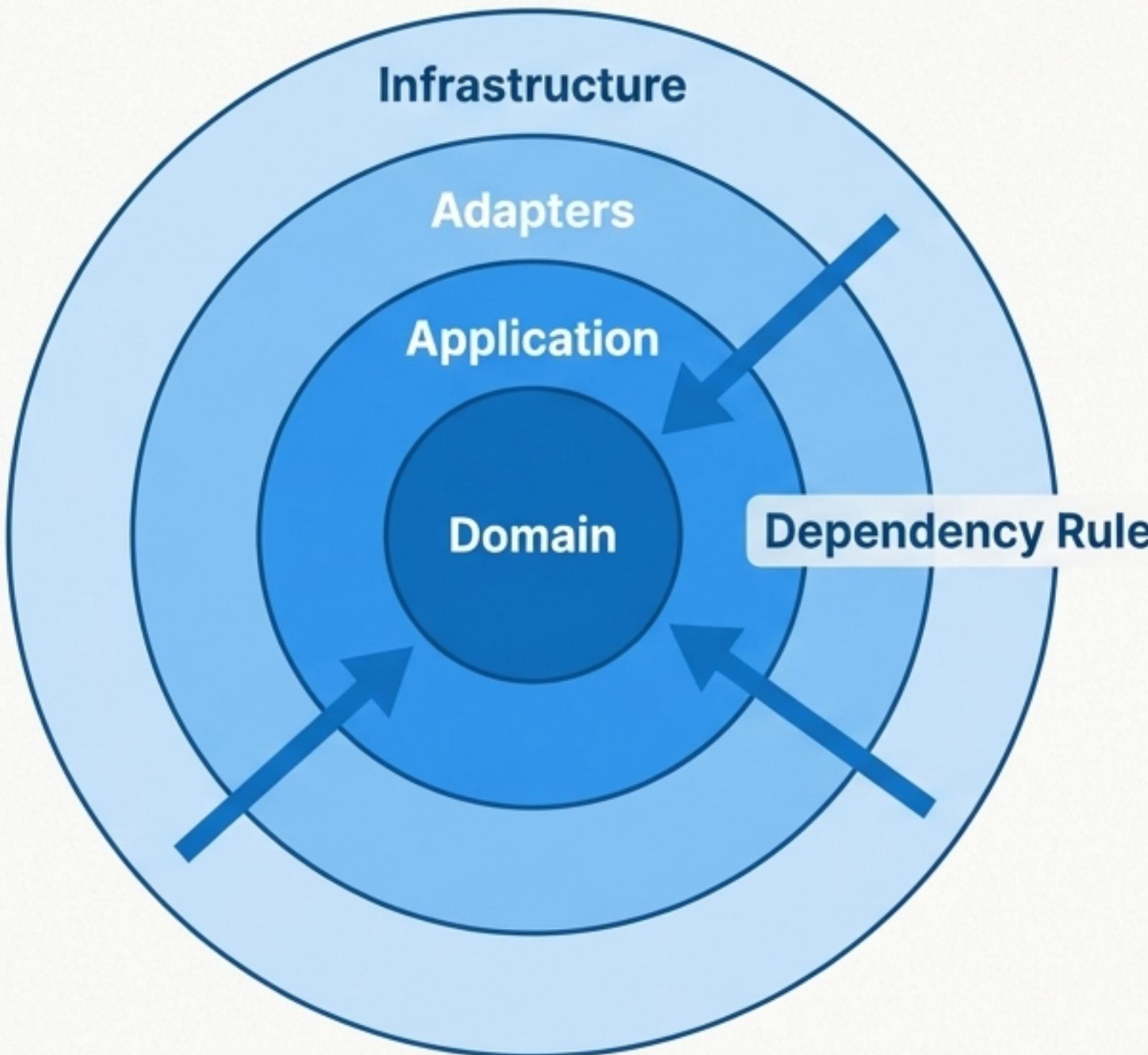
'React 컴포넌트' 요청 시 `react.md`의 베스트 프랙티스를 자동 적용합니다.

# AI의 '건망증' 해결: 중단 없는 컨텍스트



- **자동 저장/로드:** 세션 시작 시 이전 상태를 자동으로 로드하고, 종료 시 현재 상태를 저장합니다.
- **Context Compact:** 대화가 길어져도 핵심 정보(규칙, 작업 상태)는 `CURRENT\_CONTEXT.md`에 영구 보존됩니다.
- **수동 복원:** `/restore-context` 명령어로 언제든지 핵심 컨텍스트를 다시 불러올 수 있습니다.

# 견고한 설계를 위한 클린 아키텍처 강제



- `/clean-init` : 4-레이어 디렉토리 구조를 즉시 생성합니다.
- `/clean-entity User` : 순수 비즈니스 로직을 담는 도메인 엔티티를 생성합니다.
- `/clean-usecase CreateUser` : 애플리케이션 유스케이스를 템플릿에 맞춰 생성합니다.
- `/clean-validate` : 전체 코드의 의존성 규칙 위반 여부를 검증하고 보고서를 제출합니다.

# 사용 시나리오: 신규 기능 개발



**PM:** JIRA에 '사용자 인증' Epic 생성

**개발자:** `/dev plan 사용자 인증 시스템` → PRD 자동 생성

**개발자:** `/dev design` → 아키텍처, ERD 설계

**개발자:** `/dev tasks` → Worktree에 구현 태스크 자동 생성

**개발자:** `/jira-push` → Worktree의 태스크들이 JIRA 이슈로 자동 생성

**개발자:** `/dev build TASK-001` → 구현 시작 (JIRA 상태 'In Progress'로 자동 변경)

**개발자:** 작업 완료 → JIRA 상태 'Done'으로 자동 변경

---

**\*\*결과\*\*:** 체계적인 프로세스를 통해 일관된 품질과 투명한 진행 상황 공유를 모두 달성합니다.

# 숫자로 증명되는 생산성과 품질 향상

**50%**

**온보딩 시간 단축**

(프로젝트 자동 분석)

**정성적 효과**

✓ 일관된 코드 품질 및 아키텍처

**70%**

**코드 리뷰 지적 사항 감소**

(자동 품질 검사 및 패턴 적용)

✓ 체계적인 문서화 문화 정착

**90%**

**컨텍스트 재설명 시간 절감**

(자동 상태 복원)

✓ 기술 부채 발생 억제

# 지금 바로 시작하세요

## 1단계: 설치 (1분)

```
git clone [repository_url]  
cp -r .claude /path/to/your-project/
```

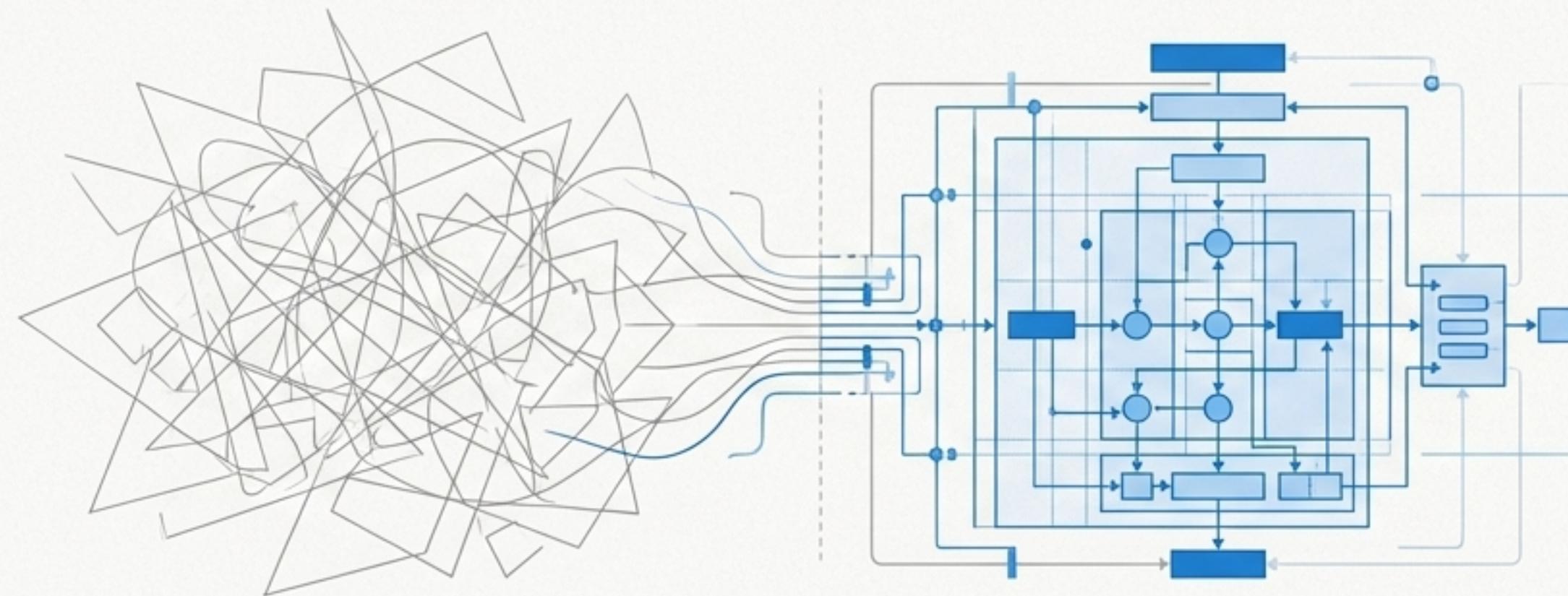
## 2단계: JIRA 연동 (선택 사항)

```
export JIRA_API_TOKEN='your-api-token'  
/jira-init YOUR_PROJECT_KEY
```

## 3단계: 첫 명령어 실행

- 새 프로젝트: `/dev plan 'My new amazing feature'`
- 기존 프로젝트: `/onboard`

# AI 개발의 혼돈에서 통제된 일관성으로



원더 무브 연구소 플러그인은 개발자의 창의성을 극대화하고, 시스템이 품질을 보장합니다.