「개발 프로세스 개선 사례」

PM이 개발을 기다리던 팀, 이제는 개발이 PM을 기다립니다

AI에 공학의 흐름을 접목한 A-RMS 팀

활용한 AI 도구 Claude Code (Claude Opus · Claude Sonnet). 처음에는 Claude Code · Codex · Gemini를 함께 쓰는 멀티 벤더 방식으로 시작했지만, 벤더마다 따로 늘어나는 결제와 관리가 부담이 됐습니다. 돌아보면 그 과정은 지금의 멀티 모델 방식에 이르는 길이었습니다. 지금은 Claude Code 세션이 작업을 지휘하고, 작성은 Claude Opus가, 검토는 Claude Sonnet이 맡습니다.

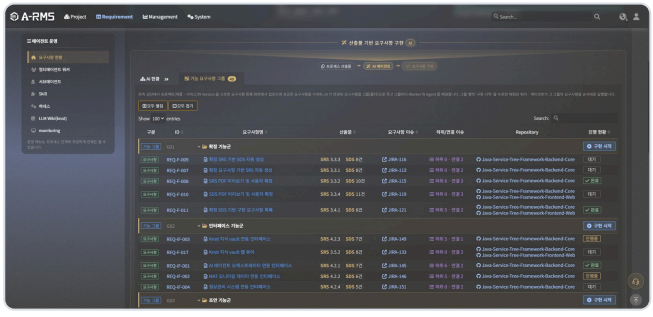
한 줄 소개 AI에게 바로 코드를 시키지 않고 공학의 흐름을 접목했습니다(온고지신). AI가 산출물을 만들고 사람이 검토해, 완성도 있는 산출물이 다음 산출물의 입력이 되게 했고, 마지막에는 그 근거 문서로 요구사항을 실행하게 했습니다. **공학은 옳았습니다.**

백문이 불여일견 https://www.a-rms.net | Id : demo | Password : Arms@313devgrp

AI Agent 허브



AI Agent 허브 · 프로세스 산출물 관리



산출물 기반 요구사항 구현

프로젝트 관리 산출물 + 개발 프로세스 산출물 : 착수 → 분석 → 설계 → 구현 → 시험



관리 산출물 · PMBOK



착수 · 배경 자료



분석 · SRS



설계 · SDS



구현 · TASK



시험 · TC

에이전트 운영



멀티에이전트 워커



서브에이전트



Skill



하네스



LLM Wiki(knot)



monitoring

① 어떤 상황에서 AI를 활용했나요?

과거에는 PM이 개발을 기다렸습니다

저희 팀은 AI 기반 프로젝트 · 요구사항 관리 시스템인 A-RMS를 만들고 운영합니다. A-RMS는 제안요청서나 기획서에 적힌 요구사항을 Jira · Redmine 같은 이슈 트래커의 이슈로 배포해 연결합니다. 이후 요구사항 단위로 이슈를 수집하고, 일정 · 범위 · 인력 · 비용을 추적합니다. 여기에 더해, 'AI Agent 허브'라는 이름의 시제품을 만들고 있습니다. 수집된 자료로 프로젝트 관리 산출물과 개발 프로세스 산출물을 구성해 늘 최신으로 유지합니다. AI 에이전트는 이 산출물을 근거로 요구사항 단위 개발을 수행합니다.

✓ 과거에는 PM이 개발을 기다렸습니다.

- 기획자는 문서와 PPT로, PM은 엑셀로, 개발자는 이슈 트래커로 일하는 경우가 많습니다. 요구사항과 실제 이슈 사이에 연결이 없으니, 관리자는 요구사항이 아니라 이슈 개수로 진척을 셉니다.
- 개발은 흔히 가장 오래 걸리는 구간입니다. PM은 결과물이 나올 때까지 기다렸다가, 그제야 전달된 요구사항과 이해된 요구사항이 달랐다는 것을 알게 됩니다. 이렇게 구현 · 운영 단계에서 고치면 비용이 요구사항 단계보다 **10~100배**까지 커진다고 알려져 있습니다(IREB CPRE, 원전 B. Boehm 1981).
- A-RMS는 이 끊어진 연결을 잇기 위해 만든 시스템입니다.

✓ 코딩 에이전트를 쓰자 기다리는 쪽이 바뀌었습니다.

A-RMS를 개발하면서 저희도 코딩 에이전트를 본격적으로 썼습니다. 코드가 나오는 속도가 빨라지자 병목은 개발이 아니라 개발에 넣을 입력, 곧 요구사항과 산출물로 옮겨 갔습니다. 근거 문서 없이 일을 맡기면 에이전트는 빈칸을 그럴듯한 추측으로 채웠습니다.

- 요구사항에 없는 기능이 들어갔습니다.
- 모듈 구조나 기존 규약과 맞지 않는 코드가 나왔습니다.
- 있지도 않은 기능을 있는 것처럼 썼습니다.

좋은 입력만 있으면 개발은 금방 끝납니다. **이제는 개발이 PM의 산출물을 기다립니다.**

✓ 그래서 공학의 흐름을 AI에 접목했습니다.

새 방법을 만들지 않고, 사람에게 써 온 공학의 순서를 AI에게 그대로 적용했습니다. 산출물을 먼저 만들고, 사람이 검토해 완성도를 높인 뒤, 그 산출물을 다음 산출물과 개발의 입력으로 씁니다. 저희는 2026년 9월부터 이 방식으로 A-RMS 개발 저장소에서 실제 작업을 하고 있고, 같은 흐름을 A-RMS 안으로 옮긴 것이 AI Agent 허브 시제품입니다.

② 어떤 AI를 어떻게 활용했나요?

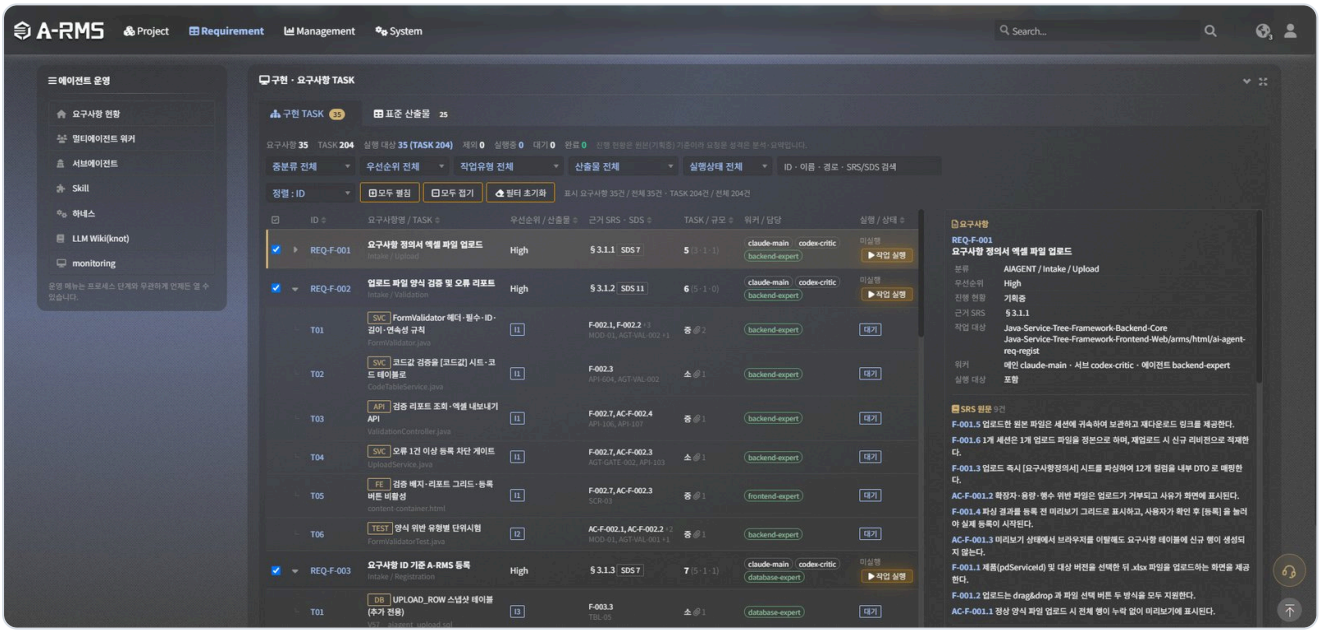
AI가 만들고, 사람이 검토하고, 다음 입력이 됩니다

AI 활용은 두 층으로 나뉩니다. 하나는 팀이 실제로 A-RMS를 개발하는 방식이고, 다른 하나는 그 방식을 A-RMS 화면으로 옮기는 AI Agent 허브입니다.

✓ 1) 산출물의 사슬: AI가 만들고, 사람이 검토하고, 다음 입력이 됩니다

- 작업마다 폴더를 만들고 네 가지를 파일로 남깁니다. 목표 · 제약을 적은 작업 정의, 에이전트에게 주는 지시서, 에이전트의 결과, 작업 로그입니다.
- 지시서에는 내용을 붙여 넣지 않고 근거 문서의 경로만 적습니다. 에이전트는 요구사항 명세서 · 설계서 같은 프로젝트 산출물을 직접 읽고 그 범위 안에서 일합니다.
- 에이전트가 만든 결과는 검토와 사람의 확인을 거친 뒤 다음 작업의 근거 문서가 됩니다. **앞 산출물의 완성도가 다음 산출물의 품질을 정합니다.**

허브 시제품 화면



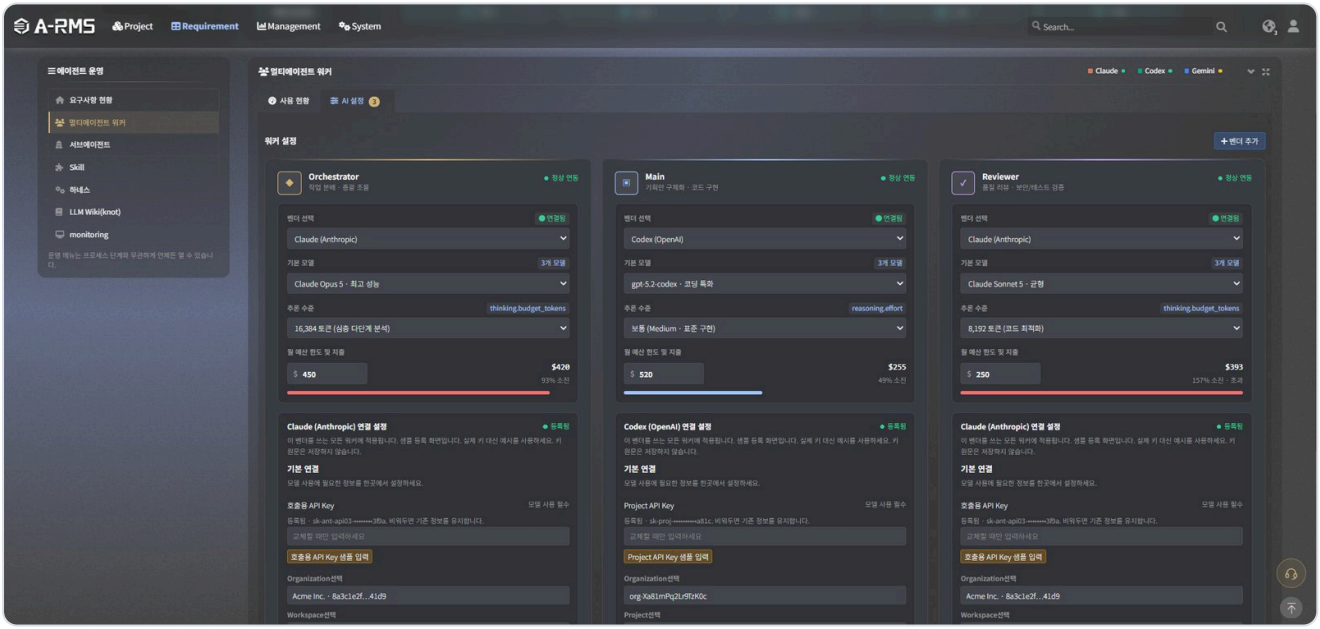
▲ 구현 · 요구사항 TASK — 요구사항마다 근거 문서(SRS · SDS)와 작업 단위를 나란히 둡니다. 에이전트는 이 근거 범위 안에서 일합니다

② 어떤 AI를 어떻게 활용했나요?

역할 분리: 멀티 벤더에서 멀티 모델로

- Claude Code 세션이 작업을 나누고 지휘하는 오케스트레이터를 맡습니다. 작성 · 설계는 **Claude Opus** 에이전트가, 검토는 다른 모델인 **Claude Sonnet**의 읽기 전용 에이전트가 합니다.
- 처음에는 Codex를 다른 벤더의 교차 검토자로, Gemini를 보조로 함께 썼습니다. 벤더가 늘수록 결제와 관리가 복잡해져, 2026년 10월부터는 한 벤더 안에서 모델을 나눠 검토의 독립성을 지키는 방식으로 옮겼습니다.
- 검토는 분야별로 나눴습니다. 착수 전 계획의 허점을 공격하는 에이전트, 모듈 간 연동 계약을 대조하는 에이전트, 보안 감사 에이전트가 따로 있습니다.
- 에이전트를 부르기 전에는 **승인 게이트 스크립트**가 사람의 승인 기록과 쓰기 범위를 검사합니다. 작업이 끝난 뒤에는 허용 범위 밖의 파일이 바뀌지 않았는지 검사합니다.
- 모듈별 코딩 규약은 에이전트가 읽을 수 있는 형태로 저장소에 두었습니다. 하네스 문서 5개 모듈, 스킬 9종, 도메인 전문 서브에이전트 14종입니다. 서적 · 표준 35건을 정리한 지식 위키(knot)도 작업 전에 조회합니다.

허브 시제품 화면



▲ 멀티에이전트 워커 — 오케스트레이터 · 작성 · 검토 역할마다 AI 벤더와 모델을 고르는 설정 화면

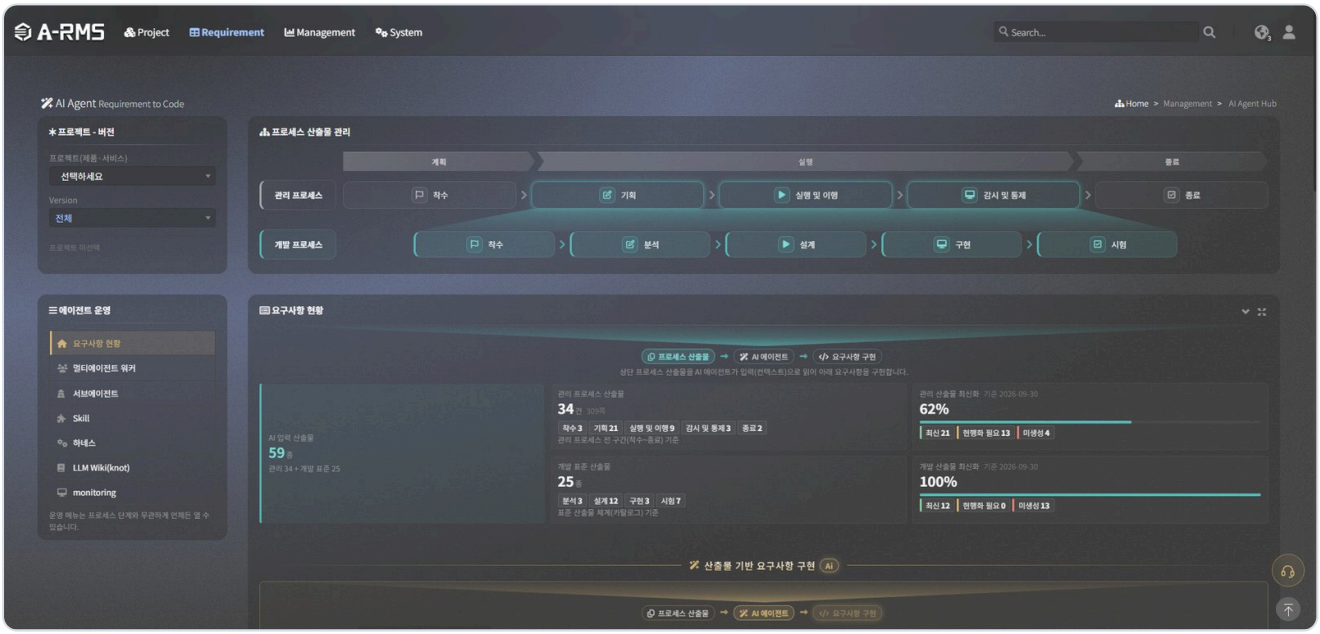
② 3) AI Agent 허브

AI를 활용한 개발 방식을 A-RMS 안으로 옮깁니다

허브는 A-RMS의 Management > AI Agent Hub 메뉴에 들어간 화면입니다.

- 산출물을 관리 5단계(착수 · 기획 · 실행 및 이행 · 감시 및 통제 · 종료)와 개발 5단계(착수 · 분석 · 설계 · 구현 · 시험)로 정리했습니다. 기준은 PMBOK과 ISO 21500입니다.
- 화면 흐름은 **'프로세스 산출물 → AI 에이전트 → 요구사항 구현'**입니다. 배경 자료 웹하드가 있고, 산출물 화면에서는 근거 문서(SRS · SDS)를 옆에 띄울 수 있으며, 요구사항 편집 화면에는 AI 수정 · AI 검증 버튼이 있습니다.
- 오케스트레이터 · 작성 · 검토 역할마다 AI 벤더와 모델을 고르는 설정 화면과, 구현을 시작하기 전에 사람이 확인하는 단계를 화면으로 구성했습니다.
- 산출물 현행화와 요구사항 기반 개발 실행은 현재 화면 흐름만 갖춘 **시제품 기능**입니다. 실제 AI 호출, 산출물 자동 생성, Jira 이슈 연결은 다음 단계 과제입니다.

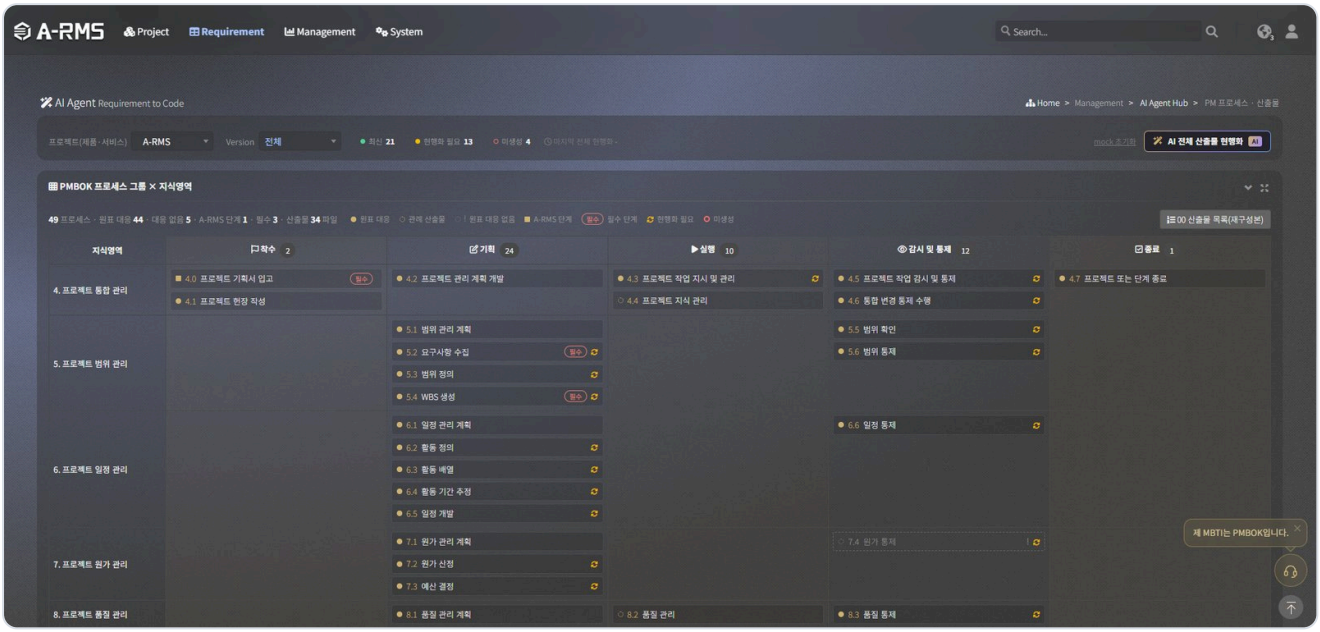
허브 시제품 화면



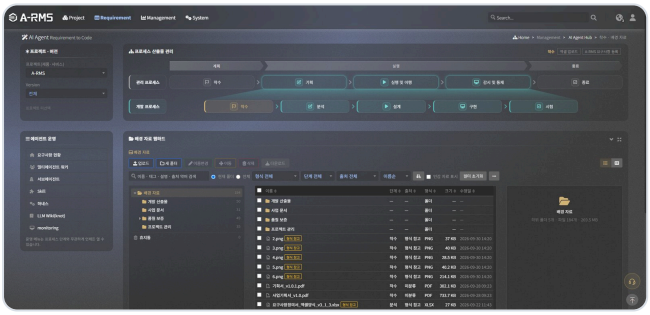
▲ AI Agent 허브 홈 — 관리 5단계 · 개발 5단계 프로세스 산출물 관리와 '프로세스 산출물 → AI 에이전트 → 요구사항 구현' 흐름

② 3) AI Agent 허브 · 시제품 화면

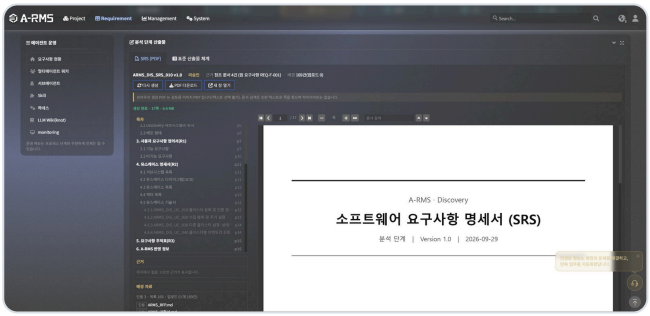
프로세스 산출물 · 관리와 개발



▲ 관리 산출물 — PMBOK 프로세스 그룹(착수 · 기획 · 실행 · 감시 및 통제 · 종료) × 지식영역



▲ 개발 1 착수 · 배경 자료 웹하드

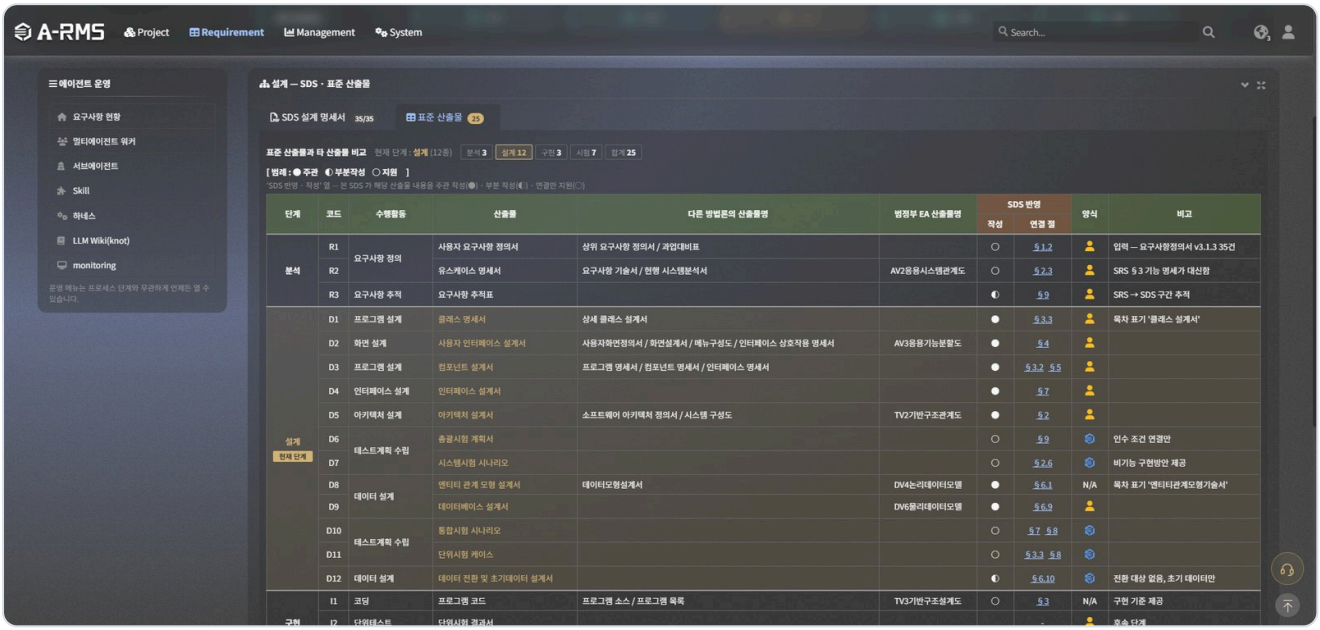


▲ 개발 2 분석 · SRS — 근거 문서를 옆에 띄워 봅니다

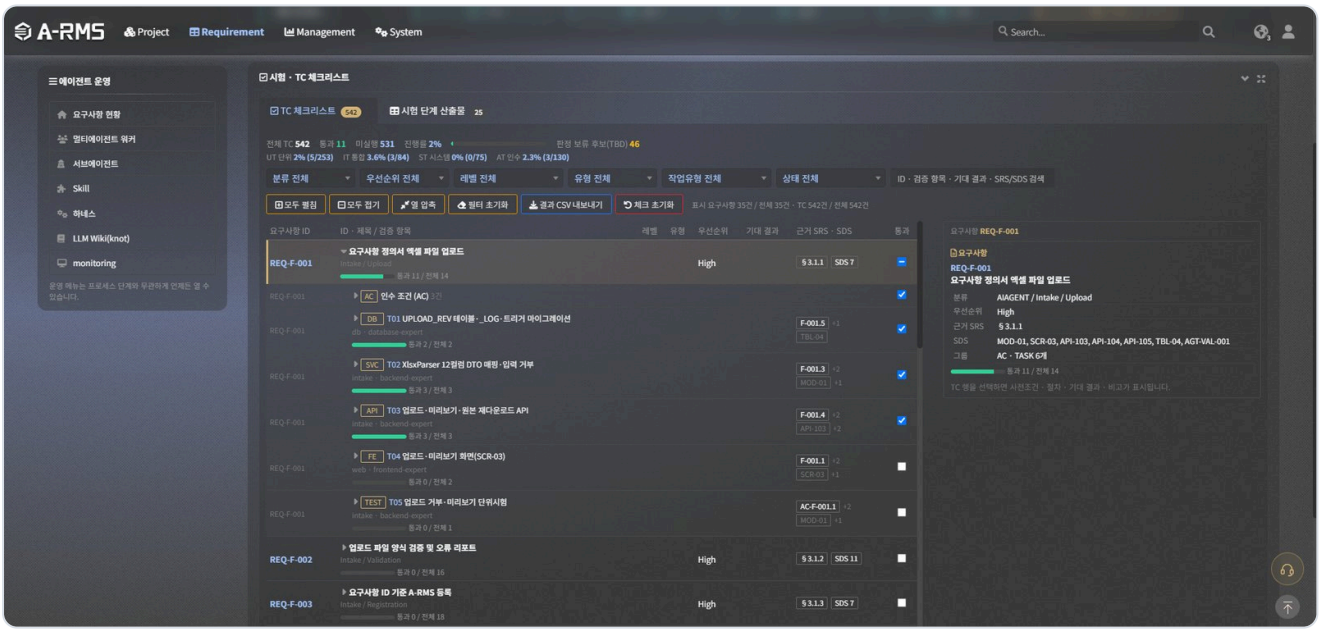
산출물 현행화와 요구사항 기반 개발 실행은 현재 화면 흐름만 갖춘 시제품 기능입니다. 실제 AI 호출, 산출물 자동 생성, Jira 이슈 연결은 다음 단계 과제입니다. 화면 속 수치 · 이슈 번호 · 상태는 화면 흐름을 보여 주기 위한 예시 값입니다.

② 3) AI Agent 허브 · 시제품 화면

개발 산출물 · 설계와 시험



▲ 개발 3 설계 · SDS와 표준 산출물을 비교해 봅니다

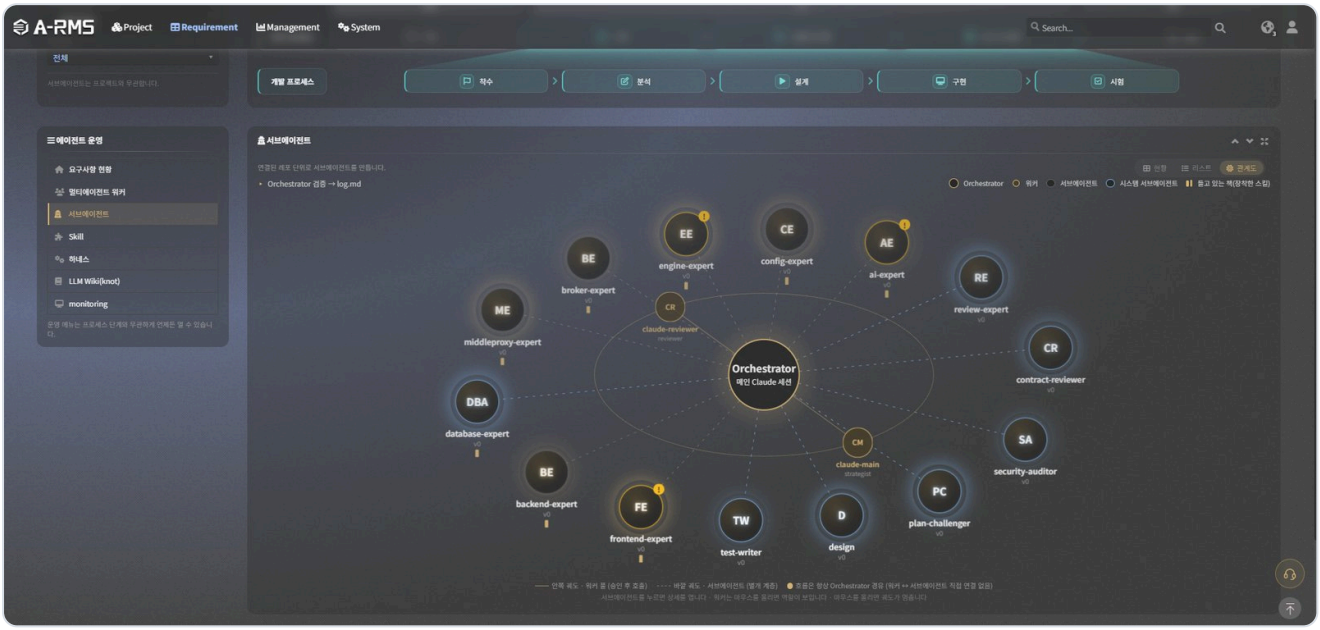


▲ 개발 5 시험 · 요구사항별 TC 체크리스트 (개발 4 구현 화면은 3쪽)

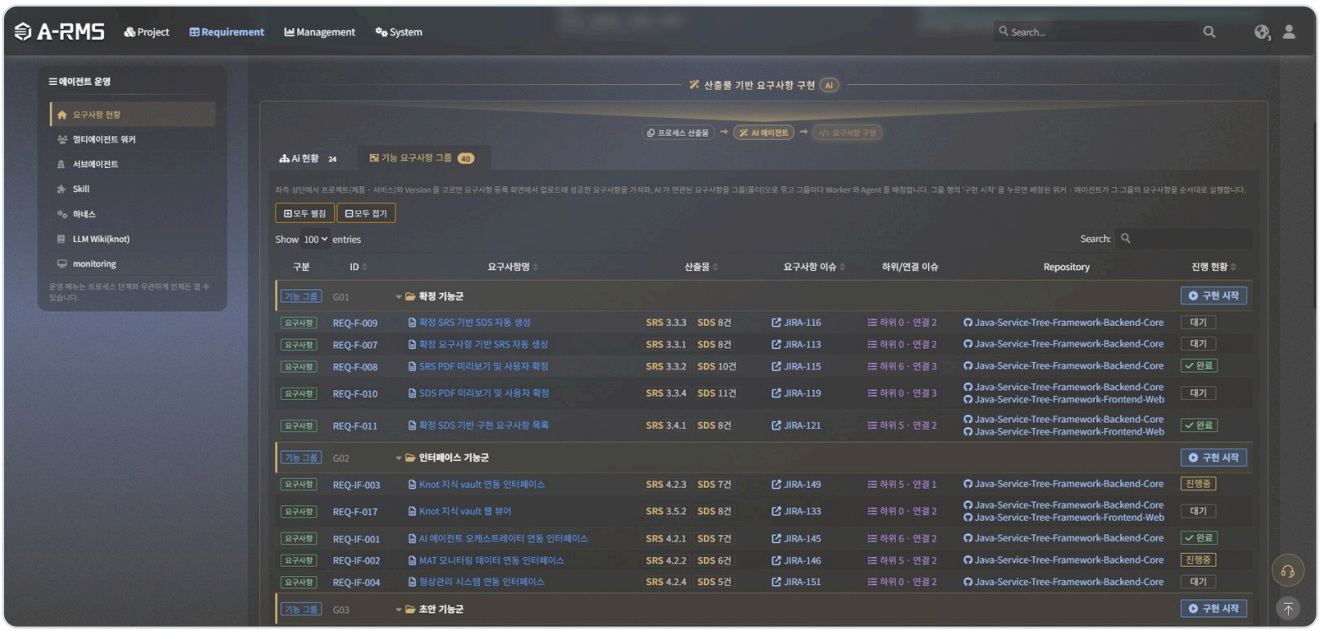
산출물 현행화와 요구사항 기반 개발 실행은 현재 화면 흐름만 갖춘 시제품 기능입니다. 실제 AI 호출, 산출물 자동 생성, Jira 이슈 연결은 다음 단계 과제입니다. 화면 속 수치 · 이슈 번호 · 상태는 화면 흐름을 보여 주기 위한 예시 값입니다.

② 3) AI Agent 허브 · 시제품 화면

AI 에이전트 → 요구사항 구현



▲ 서브에이전트 — 오케스트레이터를 중심으로 작성(claude-main) · 검토(claude-reviewer)와 계획 공격(plan-challenger) · 계약 대조(contract-reviewer) · 보안 감사(security-auditor) 등 도메인 전문 서브에이전트 14종



▲ 허브 홈 · 산출물 기반 요구사항 구현 — 요구사항 그룹마다 [구현 시작] 버튼

산출물 현행화와 요구사항 기반 개발 실행은 현재 화면 흐름만 갖춘 시제품 기능입니다. 실제 AI 호출, 산출물 자동 생성, Jira 이슈 연결은 다음 단계 과제입니다. 화면 속 수치 · 이슈 번호 · 상태는 화면 흐름을 보여 주기 위한 예시 값입니다.

③ 활용 결과 어떤 변화가 있었나요?

기다리는 쪽이 바뀌었습니다

팀이 체감한 가장 큰 변화는 병목의 위치입니다. 근거 문서가 갖춰진 작업은 에이전트가 바로 진행했고, 근거가 비어 있던 작업은 입력부터 다시 채워야 했습니다. 그래서 사람의 시간은 코드를 기다리는 데가 아니라 **산출물을 다듬고 검토하는 데** 쓰입니다.

78	253	16 / 40
약 5주간 진행한 작업 (마친 작업 54)	에이전트 호출 단위 (지시서 219 · 결과 215)	판정이 기록된 검토 결과 중 수정 요구(REVISE · BLOCKED)

✓ AI가 한 일의 출처를 따라갈 수 있게 됐습니다.

2026년 9월 23일부터 10월 29일까지 약 5주 동안 78개 작업을 이 방식으로 진행했습니다. 그중 54개를 마쳤고, 나머지는 진행 중이거나 보류 중입니다. 모든 작업에 작업 정의와 결정 로그가 남아 있고, 에이전트 호출 단위 253건 중 지시서가 남은 것은 219건, 결과가 남은 것은 215건입니다. 그래서 어떤 변경이 어떤 요구와 근거에서 나왔는지 거꾸로 따라갈 수 있습니다. 요구공학에서 말하는 **후방 추적성**(이 결과는 어디서 왔나)을 AI 작업에도 적용한 것입니다.

✓ 검토가 개인의 습관이 아니라 작업 구조가 됐습니다.

같은 기간에 작성 에이전트가 쓰인 작업은 64개, 별도 검토 에이전트가 붙은 작업은 40개, 다른 벤더(Codex)의 교차 검토가 붙은 작업은 26개였습니다. 착수 전 계획 공격, 모듈 간 계약 대조, 보안 감사 에이전트가 쓰인 작업은 각각 22개 · 7개 · 7개였습니다. 판정이 기록된 검토 결과 40건 중 16건은 수정 요구(REVISE · BLOCKED)였습니다. 근거와 어긋난 산출물이 다음 작업의 입력이 되기 전에 걸러진 것입니다. 오케스트레이터는 결과를 받아들이기 전에 지시한 형식과 맞는지, 참조한 파일이 실제로 있는지, 금지 사항을 어기지 않았는지를 점검하고 그 결과를 로그에 남깁니다.

✓ 다음 목표는 AI 작업의 정량 측정입니다.

속도는 아직 숫자로 재지 않았습니다. 먼저 확인한 것은 결과가 어디서 왔는지 따라갈 수 있느냐였습니다. A-RMS는 요구사항 기준으로 사람의 일정 · 범위 · 인력 · 비용을 계산해 왔습니다. 허브의 다음 단계는 AI 에이전트를 이슈 담당자로 등록해, 같은 지표로 사람과 나란히 보는 것입니다.

④ 나만의 방식 또는 개선 포인트는 무엇인가요?

"공학은 옳았다."

흔히 쓰는 방식은 AI 챗봇이나 코딩 에이전트에 요구사항을 붙여 넣고 바로 코드를 받는 것입니다. 저희는 반대로 갔습니다. 핵심은 한 문장입니다. **"공학은 옳았다."**

✓ 첫째, 온고지신: 새 방법론 대신 검증된 공학의 순서를 썼습니다.
PMBOK과 ISO 21500의 단계와 산출물을 AI 작업의 순서로 삼았습니다. 작업 전에는 서적 · 표준 35건을 정리한 지식 위키를 조회합니다. 그래서 AI가 일반론이 아니라 출처 있는 기준으로 판단합니다.

✓ 둘째, 입력의 완성도를 관리합니다.
AI가 만든 산출물을 사람이 검토해 다듬은 뒤에야 다음 산출물의 입력으로 씁니다. 에이전트에게는 그 근거 문서의 경로만 줍니다. 결과의 품질을 결과 단계에서 고치지 않고, 입력 단계에서 결정합니다.

✓ 셋째, 요구사항 관리 시스템의 추적 개념을 AI 작업에 그대로 가져왔습니다.
A-RMS가 요구사항과 이슈를 잇듯이, AI 작업 하나하나를 목표 · 근거 · 결과 · 검토 · 로그를 묶은 추적 단위로 남깁니다. 로그는 덧붙이기만 하는 규칙이라 결정이 언제, 왜 내려졌는지가 그대로 남습니다.

✓ 넷째, 만드는 AI와 검토하는 AI를 모델까지 갈랐습니다.
자기 결과를 자기가 채점하는 AI는 믿기 어렵습니다. 검토 에이전트는 작성 에이전트와 다른 모델이고, 파일을 고칠 수 없습니다. 멀티 벤더를 거쳐 멀티 모델로 정리하면서, 결제와 관리는 단순해지고 검토의 독립성은 그대로 지켰습니다. 코딩 규약과 작업 요령, 승인 규칙은 사람의 기억이 아니라 저장소와 스크립트에 있습니다. 그래서 AI 모델이 바뀌어도, 사람이 바뀌어도 같은 기준으로 일합니다.

✓ 다섯째, AI가 일하되 결정과 책임은 사람이 집니다.

- 사람이 승인하기 전에는 에이전트를 부르지 않습니다. 승인 기록이 없으면 승인 게이트 스크립트가 호출을 막고, 검토 AI는 파일을 고칠 수 없습니다.
- 에이전트에게는 근거 문서의 경로만 주고, 근거가 없는 값은 추측으로 채우지 않고 비워 두거나 추정이라고 표시하게 합니다. 오케스트레이터는 참조한 파일이 실제로 있는지 확인한 뒤에만 결과를 받아들입니다.
- 산출물에는 사람을 이름이 아니라 역할로 적고, 비밀번호 · 키 같은 자격증명은 에이전트에게 넘기지 않습니다.
- 지식 위키의 서적 · 표준 35건은 출처와 함께 정리했고, 이 글의 수치는 작업 로그에서 집계한 값입니다. A-RMS는 설치형이라 데이터가 조직 밖으로 나가는 범위를 조직이 정합니다.

⑤ 다른 사람도 따라 할 수 있나요?

네. 허브가 없어도 지금 따라 할 수 있습니다

필요한 것은 코딩 에이전트와 저장소 하나입니다.

1. 저장소에 AI용 작업 규칙 파일을 두고, 작업마다 폴더를 만들어 목표 · 제약 · 로그를 남깁니다.
2. 에이전트에게는 내용 대신 근거 문서의 경로와 '하지 말 것'을 짧게 적은 지시서를 줍니다.
3. 에이전트가 만든 산출물은 사람이 확인한 뒤에 다음 작업의 근거 문서로 씁니다.
4. 만드는 AI와 다른 모델의 검토 AI를 두고, 검토 AI는 읽기만 하게 합니다.
5. 사람이 승인하기 전에는 에이전트를 부르지 않고, 끝나면 허용 범위 밖의 변경이 없는지 검사합니다.

A-RMS는 Docker 기반 설치형으로 제공되어 폐쇄망 환경에도 설치할 수 있고, 핵심 프레임워크는 LGPL로 공개되어 있습니다.

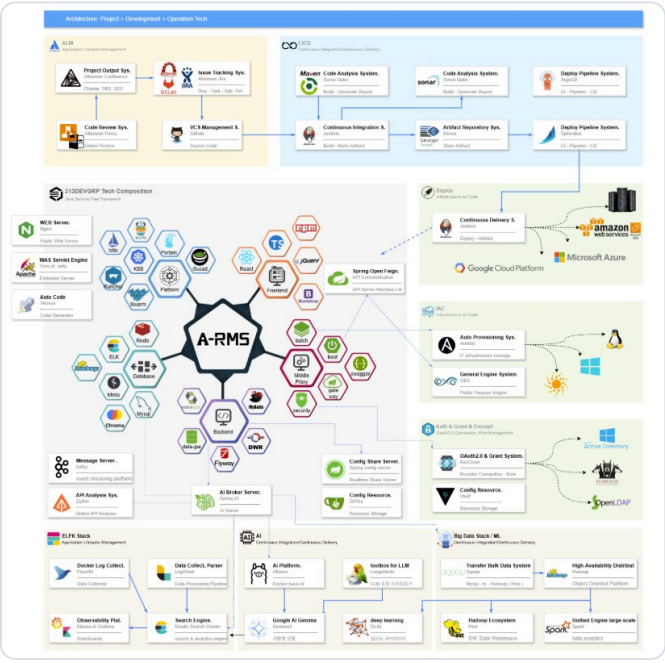
A-RMS의 요구사항–ALM 이슈 연동 및 분석 방법은 특허 출원 중입니다(출원번호 10-2025-0061451).

기반 시스템인 A-RMS(요구사항–이슈 연동)는 기존 공모전 수상작입니다. 이 사례의 대상은 그 위에서 2026년 9월부터 새로 적용한 AI 활용 개발 방식과, 그 방식을 옮긴 AI Agent 허브 시제품입니다.

백문이 불여일견 https://www.a-rms.net | Id : demo | Password : Arms@313devgrp

AI Agent 허브의 기반, A-RMS

A-RMS는 2024 대한민국 SW기술대상 및 IT 아키텍처 대전 ITA키텍처 부문 대상(과학기술정보통신부 장관상)과 2024년 공개SW 개발자대회 대상(과학기술정보통신부 장관상)을 받았습니다.



▲ A-RMS 아키텍처



▲ A-RMS 수상 이력(2024)