MSAP.ai

msap.ai

AI 시대 어떻게 일할 것인가

AI가 바꾸는 일과 개인·조직·직무별 대응

PROJECT CODE	DATE
MSAP.ai	2026.11.07

CONFIDENTIAL

일의 구조 변화와 직무별 변화

KEY INSIGHT 1부와 2부 — 일의 구조 변화와 직무별 변화

01 AI가 바꾸는 일의 구조

1.1	AI 확산 속도와 업무 영향 범위	7장
1.2	생산성 실측 결과	7장
1.3	가치가 줄어든 일과 남은 일	7장
1.4	병목의 이동	7장
1.5	일자리와 경제 전망	7장
1.6	목적 중심 업무 지시	6장
1.7	AI 활용 성과 격차의 원인	6장

02 직무별로 달라지는 일

2.1	경리·회계 사무의 변화	7장
2.2	사무직의 업무 설계 역할	7장
2.3	마케팅·HR 직무의 변화	6장
2.4	영업 직무의 변화	7장
2.5	기획·PM 직무의 변화	7장
2.6	개발자 역할의 이동	7장
2.7	개발자에게 남는 역량	7장
2.8	개발 현장의 생산성 실제	7장
2.9	디자인 업무의 변화	7장
2.10	디자이너 역할과 역량	7장
2.11	AI가 읽는 디자인 시스템	7장
2.12	테스트·QA 직무의 변화	7장

Source: 이 덱의 구성

개발 방식과 조직의 변화

KEY INSIGHT 3부와 4부 — 개발 방식의 변화와 조직이 바꿀 것

03 소프트웨어 개발 방식의 변화

3.1	에이전트에 작업을 맡기는 개발	7장
3.2	1인 개발과 작은 팀	7장
3.3	소프트웨어 팩토리 사례	7장
3.4	사람이 맡는 확인 지점	7장
3.5	하네스와 컨텍스트 설계	7장
3.6	속도가 만드는 설계·운영 부담	7장
3.7	가치 기준의 개발 생산성	7장

04 조직이 바꿔야 할 것

4.1	AI 네이티브 조직의 조건	7장
4.2	사내 AI 활용 확산	7장
4.3	AI 거버넌스와 책임	7장
4.4	팀 규모와 인력 판단	7장
4.5	AI 시대 리더의 역할	7장
4.6	AX 인재상과 채용 기준	7장
4.7	주니어 육성 공백	7장
4.8	에이전트 시대의 새 직무	6장
4.9	암묵지와 학습하는 조직	7장
4.10	손익으로 재는 AI 성과	6장

Source: 이 덱의 구성

개인의 일하는 방식과 역할별 AI 사용법

KEY INSIGHT 5부와 6부 — 개인의 일하는 방식과 역할별 사용법

05 개인이 일하는 방식

5.1	AI 시대 셀프 매니지먼트	7장
5.2	수직적 발달과 추상 역량	7장
5.3	AI 의존과 판단력	7장
5.4	변화 속도에 맞춘 학습 습관	7장
5.5	커리어 운영	6장

06 역할별 AI 사용법

6.1	AI에 일을 맡기는 기본	7장
6.2	위임 범위와 검수	7장
6.3	사무직의 AI 사용법	7장
6.4	개발자의 AI 사용법	7장
6.5	디자이너·기획자의 AI 사용법	7장
6.6	조직의 AI 도입 실행 순서	6장

Source: 이 덱의 구성

SECTION 01

AI가 바꾸는 일의 구조

AI 확산 속도·생산성 실측·병목 이동·일자리 전망·목적 정의·성과 격차

SECTION 1.1

AI 확산 속도와 업무 영향 범위

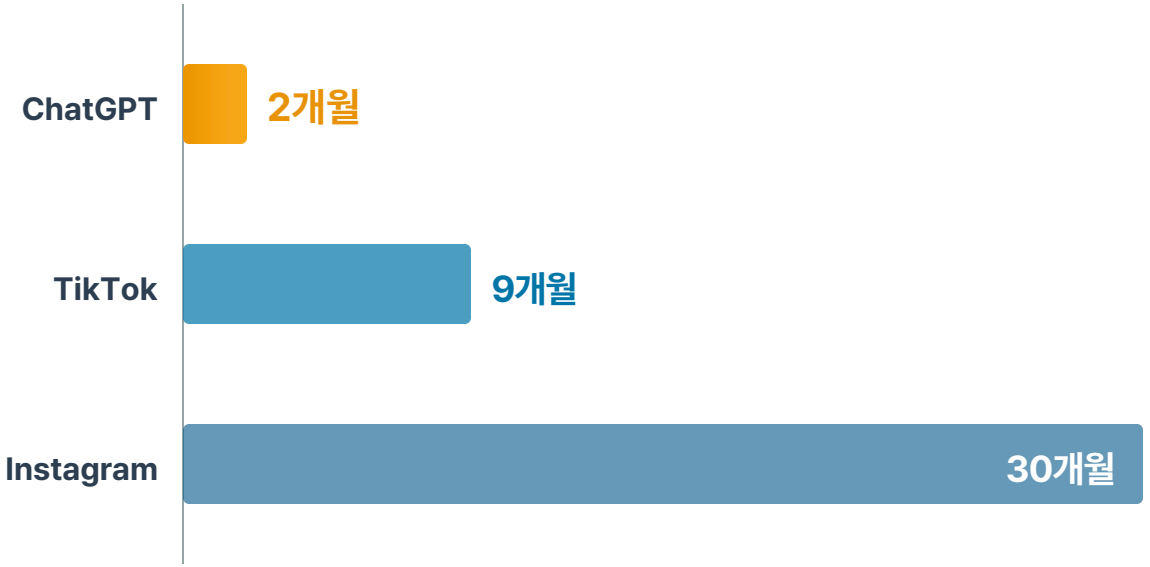
새 기술보다 빠른 확산 속도와 업무 단위로 오는 영향

1.1 월간 사용자 1억 명 도달 기간 비교

KEY INSIGHT ChatGPT가 두 달로 가장 빠르게 도달함 — 일반 사용자에게 생성형 AI 사용이 이미 일상이 됐다는 근거

월간 사용자 1억 명까지 걸린 기간(개월)

출시 후 기준, UBS 추정



같은 회사 서비스망으로 성장한 Meta Threads는 비교에서 제외(발표 자료 주석)

빌 게이츠의 평가(2023.03)

평생 혁명적이라고 느낀 기술 시연은
두 번뿐이라고 밝힘

1980년 그래픽 사용자 인터페이스(GUI)

2022년 OpenAI의 AI 모델

발표 요지

이번 AI 돌파는 역사적 전환점이며
변화 속도도 이전 기술보다 빠름

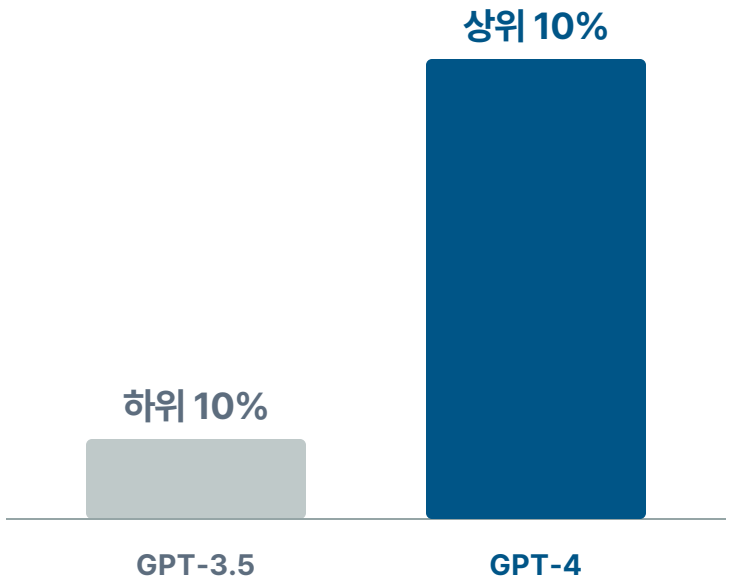
Source: 원 발표 10쪽; Reuters(2023.02.01, UBS 보고서 인용); Bill Gates, The Age of AI has begun(GatesNotes, 2023.03.21)

1.1 GPT-4의 시험 성적과 업무 수행 사례

KEY INSIGHT 법률 지식과 여러 단계 작업 모두 전문가 수준에 다가감 — 전문 직무도 AI 도입 검토 대상이 됨

미국 변호사 시험(Uniform Bar Exam)

응시자 대비 백분위 추정



보충: 응시자 구성에 따라 백분위가
더 낮게 나온다는 후속 연구도 있음

도구를 연결한 GPT-4의 신약 후보 탐색 사례

지시 한 번으로 아래 다섯 단계를 이어서 처리

1. 기존 약의 정보를 입력하고 새 후보를 찾도록 지시
2. 성질이 비슷한 화합물을 탐색
3. 특허 침해 여부를 확인하고 구조를 수정
4. 구매 가능한 공급사를 확인
5. 공급사에 보낼 주문 메일 문안까지 작성

OpenAI가 위험 평가 문서(System Card)에 공개한 시연 사례

Source: 원 발표 11쪽; OpenAI, GPT-4 Technical Report·System Card(2023); 반론 Martinez, Re-evaluating GPT-4's Bar Exam Performance(2024)

1.1 LLM의 미국 일자리 영향 추정

AI 확산 속도와 업무 영향 범위

KEY INSIGHT 영향은 고소득 직무일수록 큼 — 직무 대체보다 AI를 쓰는 사람이 쓰지 않는 사람을 대신하는 변화가 먼저 옴

업무의 10% 이상이 영향받는 노동자

약 80%

미국 노동자 전체 기준

업무의 50% 이상이 영향받는 노동자

약 19%

고소득 직무일수록 노출도가 높음

같은 품질로 훨씬 빨리 끝나는 업무 비율

47~56%

LLM 단독 15% → 기반 도구 결합 시

함께 인용된 추정

· 맥킨지: 현재 기술로 업무 시간의 60~70% 자동화 가능

· 화이트칼라·크리에이터의 직접 대체는
당분간 일어나기 어렵다는 발표 자료 판단

발표 요지: AI 파트너 역량

AI를 잘 쓰는 사람이 그렇지 않은 사람을 대신함

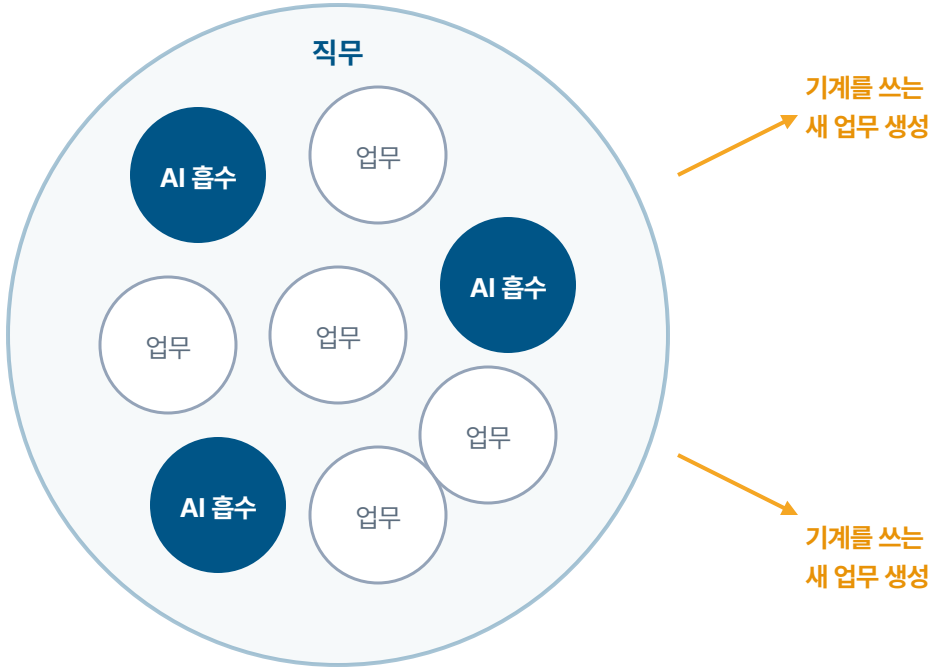
AI를 얼마나 좋은 파트너(Copilot)로 쓰는지가
인재 역량을 크게 가름

Source: 원 발표 17쪽; Eloundou 외, GPTs are GPTs(arXiv 2303.10130, 2023); McKinsey, The economic potential of generative AI(2023)

1.1 AI 영향은 업무 단위로 나타남

KEY INSIGHT Claude가 맡는 업무는 평균 교육 14.4년 수준의 고학력 업무임 — 어떤 업무가 빠지느냐에 따라 남는 직무의 수준이 갈림

하나의 직무 안에 여러 업무가 있음



직무 안의 일부 업무만 AI로 넘어감

Claude가 다루는 업무의 평균 요구 교육 연수

14.4년

경제 전체 평균 **13.2년**

미국 준학사 수준에 해당하는 고학력 업무 비중이 높음

결과는 어떤 업무가 빠지느냐에 따라 갈림

- AI는 직무 안의 고급 분석·인지 업무를 일부 흡수
- 남는 업무가 쉬우면 디스킬링, 어려우면 업스킬링
- 같은 직군 안에서도 직무마다 결과가 다름

Source: Anthropic Economic Index 2026년 1월 보고서(anthropic.com/news/anthropic-economic-index-january-2026-report), 원 자료 3쪽

왼쪽 도식은 개념 모식도이며 흡수 비율을 나타내지 않음

1.1 성과 정의가 쉬운 업무부터 AI로 이동

KEY INSIGHT 과도기에 사람이 필요한 것과 장기적으로 사람이 우위인 것은 별개 문제임 — 지금 필요하다는 사실만으로는 업무를 계속 맡을 근거가 약함

기계로 옮겨 가기 쉬운 업무의 조건

성과를 정의할 수 있음 → **결과를 비교할 수 있음** → **최적화할 수 있음**

세 조건을 갖출수록
AI로 넘어가는 속도가 빠름

발표 자료가 든 업종 — 거의 모든 산업이 같은 방향

농림수산업 · 제조업 · 건설·주거 환경 · 생활 기반 시설 · 정보통신·IT · 물류·교통 · 소매·유통 · 공중보건 · 금융·보험 · 행정 · 의료·복지 · 교육 · 사법 · 서비스 · 개발

사람의 판단이 필요해 보이는 영역도 성과를 정의하는 순간 같은 흐름에 들어감

육체노동도 방향은 같음

개선 방향: 더 빠르고, 더 정확하고, 더 높은 품질
로봇이 지금 못 하는 작업이 있어도 개선은 계속됨

지금 못 하는 일 ≠ 앞으로도 사람이 맡을 일

과도기의 필요와 장기 우위는 별개

지금 사람이 필요한 업무: 전환 기간의 공백을 메우는 일
장기적으로 남는 업무: 사람이 맡을 때 더 나은 일

두 가지를 섞으면 업무 재배치 시점을 놓침
발표 자료의 판단이며 업종별 시점은 다름

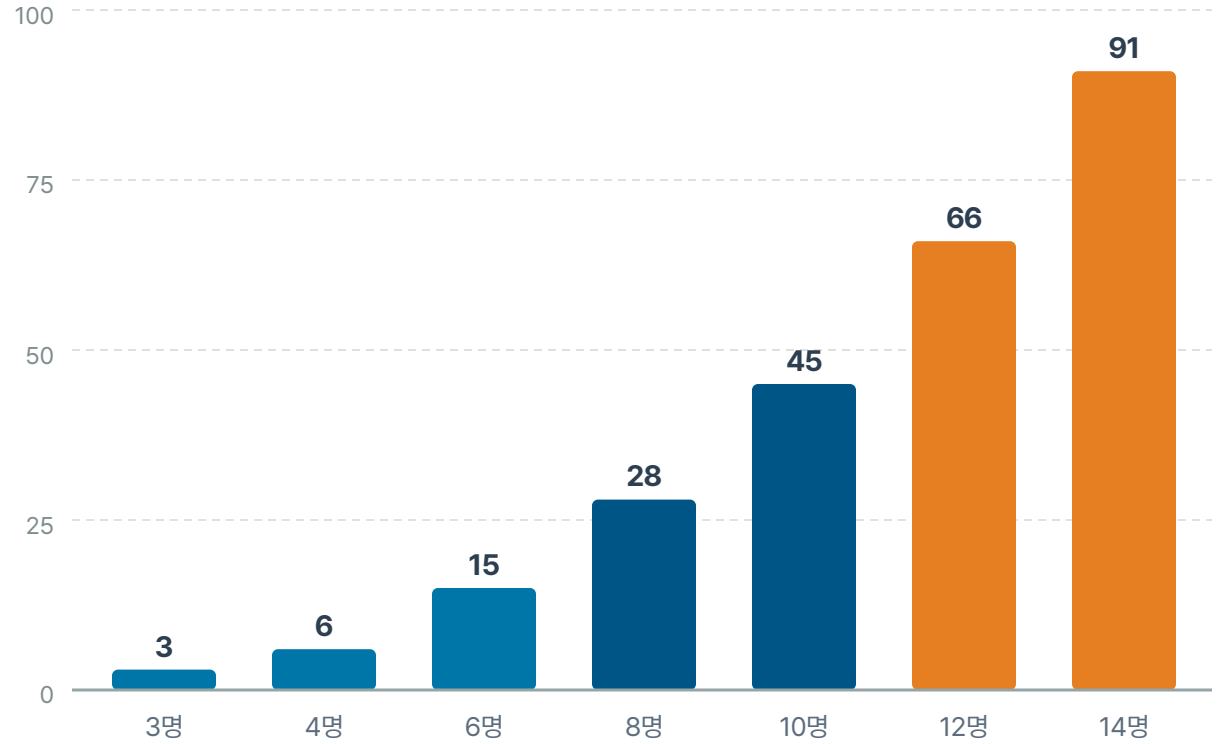
Source: 원 발표 6·9쪽(Frieve-A, 2026.08.12)

1.1 AI 대체는 아직 단순 업무에 한정됨

AI 확산 속도와 업무 영향 범위

KEY INSIGHT 참여자가 늘면 조정 경로가 n(n−1)/2로 늘어남 —
복잡한 업무일수록 완전 자동화 목표는 비현실적임

참여자 수에 따른 소통 경로 수 n(n−1)/2



14명이 되면 경로가 91개 — 3명일 때의 약 30배

AI가 사람보다 앞서는 부분

정보 처리 능력 — 속도·양·안정성

스케일링 — 병렬·비동기 처리

지식량 — 높은 품질과 다양성

데이터를 읽고 쓰며 직접 행동하는 주체로 변화

단순 업무

정해진 범위 안의
업무는 에이전트가
대신 처리 가능

복잡한 업무

의사결정 경로가
복잡해 완전
자동화가 어려움

Source: Jason Evanish, Developing Leaders: 6 Critical Things You Must Do When Your Team Grows Too Big; 원 발표 p.17~19

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

SECTION 1.2

생산성 실측 결과

BCG·상담원·METR·DORA 실측으로 본 효과의 방향

1.2 BCG 컨설턴트 758명 실험 결과

KEY INSIGHT 하위 성과자의 향상 폭이 가장 큼 — 다만 AI 능력 밖 과제에서는 정답률이 떨어져 적용 범위 판단이 필요함

GPT-4 사용 그룹과 미사용 그룹 비교(AI 능력 안쪽 과제)

완료한 과제 수

+12.2%

더 많은 과제를 끝냄

과제 처리 시간

−25.1%

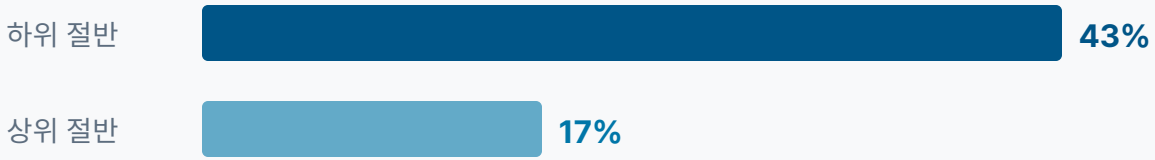
더 빨리 끝냄

결과물 품질

+40%

이상 높은 평가를 받음

기준 성과 구간별 향상 폭



발표 요지: 낮은 성과자의 역량을 끌어올리는 효과가 큼

보충: AI 능력 밖 과제의 결과

AI 사용 그룹의 정답률이
미사용 그룹보다 19%p 낮음

어떤 과제가 AI 능력 안에 드는지
먼저 가려야 효과가 남음

Source: 원 발표 18쪽; Dell'Acqua 외, Navigating the Jagged Technological Frontier(HBS Working Paper 24-013, 2023)

💡 **KEY INSIGHT** 초보 상담원의 처리량이 가장 크게 늘어남 — 우수 상담원의 응대 방식이 AI를 거쳐 초보에게 전해졌기 때문

+14%

상담원 전체 시간당 해결 건수

대화 보조 AI 도입 전후 비교

+34%

경력이 짧은 초보·저숙련 상담원

가장 큰 폭으로 늘어난 집단

≈0%

숙련 상담원

향상 폭이 거의 없음

조건: 답이 반복되는 정형 상담 과업이고, 좋은 응대의 본보기가 이미 있었음 — 판단이 필요한 과업에는 그대로 옮길 수 없음

Source: Brynjolfsson·Li·Raymond, Generative AI at Work, NBER Working Paper w31161(2023)

KEY INSIGHT AI 효과는 조직과 사람의 기존 역량에 따라 커지기도 작아지기도 함 — 도구 도입만으로 성과를 예상하면 계획이 틀어짐

METR 무작위 대조 실험 (2025)

숙련 오픈소스 개발자 16명, 실제 저장소 작업에 AI 허용 여부를 배정

본인 체감	실제 측정
20% 단축	**19% 지연**
작업 뒤 응답	AI 허용 작업의 소요 시간

빨라졌다는 느낌과 실제 결과가 반대로 나옴

산출 속도만 보고 효과를 판단하면 착시가 생김

DORA 2025 AI 보고서

Google Cloud DORA 팀의 소프트웨어 개발 조직 조사

AI는 증폭기

조직의 강점과 약점을 함께 키움

기반이 약한 조직에서는 AI가 혼란도 함께 키움

도구보다 조직의 기존 역량이 결과를 가름

PM에게 주는 시사점

성과는 산출물을 판단하는 역량에 달림 — 발표 자료가 말한 '실력 이상은 되지 않음'과 같은 결론

Source: metr.org/blog/2025-07-10-early-2025-ai-experienced-os-dev-study · dora.dev/research/ai

이 장 전체가 보충(원 발표에 없음). 수치는 각 연구 공개 자료 기준

KEY INSIGHT 개인 효과는 크지만 조직 성과는 덜 오름 — 팀과 조직이 AI 성능을 다 살리지 못하는 병목이 됨



업무에 AI를 쓰는 개발 실무자

90%

2024년 76%에서 상승

주요 용도

신규 코딩 · 기존 코드 수정 · 리뷰

문서 생성 · 테스트 케이스 생성

조사 대상 약 5,000명 (보충)

AI 효과: 개인과 조직의 차이

개인 수준

AI 활용 효과가 큼

작업 지원에서 함께 작업하거나 맡기는 단계로 넘어가는 중

조직 수준

개인만큼 효과가 오르지 않음

팀과 조직의 일하는 방식이 개인의 AI 활용을 따라가지 못함

딜리버리 처리량

소폭 증가

2024년 조사의 감소에서 반전

딜리버리 불안정성

증가

변경 실패와 재작업이 늘어남

Source: Google DORA, State of AI-assisted Software Development 2025 (원 발표 인용) · infoq.com/news/2025/09/dora-state-of-ai-in-dev-2025 (보충: 2024년 수치·응답자 수)

1.2 AI 사용자 체감과 신입 채용 감소

KEY INSIGHT AI 영향은 해고보다 경력 첫 단계의 축소로 먼저 나타남 — 이 흐름이 이어지면 장기 인력 양성이 끊길 수 있음

AI 사용자의 체감

Anthropic 사용자 조사 약 9,700명(2026.06)

86% 작업 속도가 빨라졌다는 응답

82% 업무 범위가 넓어졌다는 응답

약 10% 본인 실직 가능성이 높다는 응답

86%·82%는 원 자료 기준, 1차 출처에서 미확인

노동시장의 경고 신호

신입·초기 경력층에서 먼저 나타남

3분의 1 이상

응답자 가운데 후배·신입의 1년 내 실직 확률을 60% 이상으로 본 비율

13~14% AI 노출이 큰 직종의 22~25세 고용 상대 감소

원 자료 약 14%, 스탠퍼드 원 논문 13%(2025.08)

초기 영향의 형태

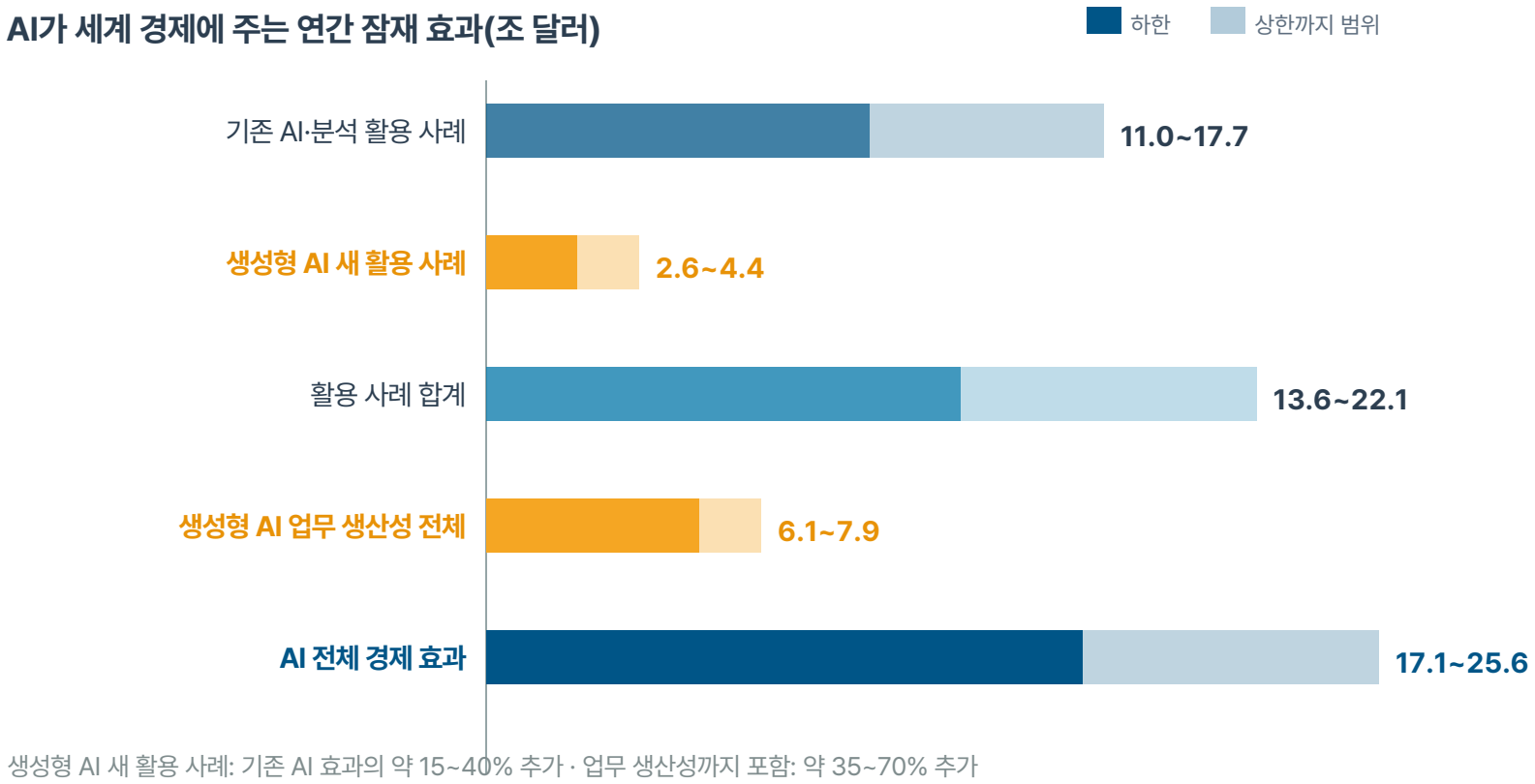
대규모 해고보다 신입이 들어오는 입구가 좁아지는 모습으로 먼저 드러남

Source: 원 자료 5쪽, Anthropic Economic Index 2026.06 보고서 보도, Brynjolfsson 외 'Canaries in the Coal Mine?'(Stanford, 2025)

사용자 조사는 Claude 개인 계정 이용자 중 응답자라 낙관 쪽으로 치우칠 수 있음(보충)

KEY INSIGHT 기존 AI 효과에 15~40%를 더하고 업무 생산성까지 넣으면 35~70%를 더함 — 경영 의제로 다룰 규모임

AI가 세계 경제에 주는 연간 잠재 효과(조 달러)



생성형 AI 새 활용 사례: 기존 AI 효과의 약 15~40% 추가 · 업무 생산성까지 포함: 약 35~70% 추가

규모 비교(보충)

생성형 AI 업무 생산성

6.1~7.9

조 달러 / 연

한국 명목 GDP(2023년)

약 1.7

조 달러

발표 자료는 일본 GDP (약 5,100조 원)과 비교하며 규모를 강조함

Source: 원 발표 21쪽; McKinsey Global Institute, The economic potential of generative AI(2023.06); 한국 GDP는 한국은행 집계(2023년 1조 7,131억 달러)

SECTION 1.3

가치가 줄어든 일과 남은 일

정형 작업의 가치 하락과 정의·판단 업무의 비중 증가

KEY INSIGHT 요리 담당자의 가치는 불을 어떻게 써야 맛있고 안전한지 아는 데 있음 — 자기 업무에서 작업과 본질을 나눠 봐야 함

발표 자료의 예: 원시 시대 요리 담당자와 라이터

이전: 불 피우기 전문가

나무 막대를 돌려 불을 피움
요령이 필요해 시간이 많이 듦
불 피우기를 자기 일로 여김

→

변화: 라이터 등장

누구나 쉽게 불을 쓰게 됨
일이 없어졌다고 낙담함
사라진 것은 불 피우기 작업뿐

→

그대로 남는 가치: 요리

불을 어떻게 써야 고기가
맛있고 안전하며 모두가
만족하는지 아는 것

발표 자료의 결론: 뺏기는 것은 일부 작업

자기 업무에 던질 두 질문

질문 1 · 작업

내 업무에서 불 피우기에 해당하는 작업은 무엇인가

보충 예: 케이스 작성, 정해진 절차 실행, 결과 집계

질문 2 · 본질

내 업무에서 요리에 해당하는 본질은 무엇인가

보충 예: 어떤 리스크를 어디서 확인할지 판단하는 일

Source: 원 발표(MIXI, 2026.07.23) 마지막 비유·당신 일의 진짜 가치 쪽

비유는 원 발표의 요지. 질문 아래 테스트 업무 예시는 보충

KEY INSIGHT 사람의 일이 만드는 일에서 정의·판단·실행으로 옮겨감 — 업무마다 사람이 개입하는 정도는 다름

사람에게 남는 일 어떻게 하고 싶은지라는 의지	AI가 맡는 일 성과물 출력, 곧 만드는 일
무엇을 왜 만들지 결정 — Why · What	데이터 분석
AI 결과물을 가려내고 판단·수정 — Go/No-Go	디자인 시안 작성, 코드 작성
주변을 움직여 끝까지 실행 — Execution	테스트에서 찾은 오류 자동 수정

AI가 만든 결과물도 사람의 Go/No-Go 판단을 거침

왼쪽 두 번째 줄의 판단·수정이 오른쪽 모든 업무의 품질을 결정함

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 16쪽

KEY INSIGHT AI가 How를 맡아도 Why·What 정의와 Go/No-Go 결정은 넘길 수 없음 — 개입할 곳과 뺄 곳을 가려야 품질과 속도를 함께 얻음



설계 지점 1

의사결정 설계

과제

AI가 How를 맡아도 Why·What 정의와 Go/No-Go 결정은 사람만 할 수 있음

설계 방향

사람이 개입할 곳과 개입하지 않을 곳을 정해 품질과 속도를 함께 확보



설계 지점 2

Context 설계

과제

Context는 시간이 지날수록 커짐. 쌓인 Context를 쓰면서도 너무 비대해지지 않게 해야 함

설계 방향

Context 양을 고려한 구조를 잡고, 키우는 방법과 활용 방법을 함께 설계



설계 지점 3

신뢰 설계

과제

AI는 틀릴 수 있고 100%는 없음. 이 전제에서 시스템 신뢰성을 계속 지켜야 함

설계 방향

틀린다는 전제로 확률에만 기대지 않는 품질·신뢰성 확보 장치를 마련

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 15쪽

원 발표자도 세 지점을 가설로 제시 · Go/No-Go: 출시 여부 결정

KEY INSIGHT 코드를 쓰는 AI는 엔지니어 결과물의 사용자이기도 함 — 좋은 아이디어를 빠르게 구현할 수단이 늘어남

가치가 줄어든 일

코드를 쓰기만 하는 일

무엇을 만들지 정해진 코드를 옮겨 적는 일은
AI가 더 빠르게 처리

차별점으로 삼기 어려움

늘어난 가능성과 남은 가치

아이디어를 더 빨리 형태로 만듦

시제품까지 걸리는 시간이 짧아짐

좋은 아이디어는 여전히 희소함

무엇을 만들지는 사람이 정함

엔지니어 가치의 중심이 이동

발표 요지

코드를 쓰는 AI는 일을 빼앗는 존재보다 엔지니어 결과물의 사용자에 가까움

AI가 쓰기 좋은 라이브러리·도구·문서를 만드는 것도 엔지니어의 일

Source: 원 발표 22쪽(uhyo, 카오나비, 2026.06.18). 카드 안 설명 줄과 하단 둘째 줄은 보충

KEY INSIGHT AI는 장당 단가 1/300, 월 처리량 약 660배임 — 채용 없이 처리량을 늘릴 수 있어 업무 설계 기준이 달라짐

같은 요청: 종이 서류 내용을 Excel에 입력

사람에게 맡김

처리 시간	장당 1시간
단가	시급 약 12,900원
월 처리 가능량	196장

늘리려면 채용 필요 · 확장 어려움

월 처리량
약 660배

장당 단가
1/300

AI에 맡김

처리 시간	장당 20초
단가	장당 약 43원
월 처리 가능량	129,600장

채용 없이 확장 가능

발표 요지

사람에게 맡기면 월 약 260만~430만 원 이상이 듦 — 하루 8시간, 22일 일해 그 돈을 받을 만한 일만 업무로 보는 편향이 생김

사람도 실수하며, AI에 맡길 수 있는 일은 아직 한정적이지만 그 범위가 넓어지는 중

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 20쪽. 수치는 원 발표의 계산 예시를 원으로 환산(1엔≈8.58원, 2026-11-08)

KEY INSIGHT 사람 단가로 맡길 수 있던 일은 맡기고 싶은 일의 일부였음 — AI 단가에서는 탐색할 수 있는 업무 범위가 크게 넓어짐

AI 시대 업무 범위

맡기고 싶은 일 전체

사업에서 맡기고 싶은 일은 끝이 없음

비용상 AI에 맡길 수 있는 일

비용 제약이 풀려 탐색 범위가 한꺼번에 넓어짐

비용상 맡길 수 있던 일 (사람 단가 기준)

인건비가 높아 전체의 아주 일부였음

지금의 범위



기존 신규 사업 관점 — 성능 기준

- 드러난 과제를 지금보다 쉽게 해결 (SaaS 등)
- 인건비를 줄여 주므로 비용을 받기 쉬움
- 사는 쪽도 과제를 알고 있어 단기간에 팔림
- 투자금 회수 기간과 잘 맞았음

발표 자료의 표현: 반걸음 앞선 사업

새 관점 — 비용 대비 효과 기준

- 비용 때문에 손대지 못한 잠재 과제에 AI 적용
- 사람에게 맡긴다는 범위로만 생각하지 않음
- 사람과 일하는 데 익숙해 비용 필터를 늘 거는 습관에 주의



Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 19·21쪽

SECTION 1.4

병목의 이동

구현이 빨라진 뒤 병목이 옮겨 가는 자리

KEY INSIGHT 사양 이해·설계가 앞으로 당겨지고 리뷰가 두 번 생김 — 줄어든 작성 시간을 다른 단계가 가져감

기대한 변화

도입 전 예상

AI 이전 | 사양 이해 · 설계 · 구현 · 리뷰

AI 이후 | 프롬프트로 작업 효율화 → **리드타임 단축**

AI가 작업을 효율화해 리드타임이 짧아질 것으로 예상

실제 변화

발표 자료자 관찰



- AI에 지시하려면 사양 이해와 설계를 미리 충분히 해 둬야 함
- AI 결과를 담당자가 먼저 리뷰하고, 그 결과를 팀이 다시 리뷰함
- 결과적으로 **리드타임은 크게 달라지지 않음**

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07). 막대 길이는 개념 표시이며 측정값이 아님

KEY INSIGHT 팀이 함께 이해를 깊게 하고 나누는 구조가 필요함 — 개인의 AI 활용만으로는 늘어난 변경을 팀이 소화하지 못함

도구와 사람

요지: 사람이 도구를 만들고, 그 뒤에는 도구가 사람을 바꿈

John M. Culkin, 1967 (흔히 매클루언의 말로 알려짐)

만드는 속도

크게 빨라짐

AI가 코드·문서·테스트 작성을 맡음

같은 시간에 만드는 변경량이 늘어남

이해하는 속도

그대로임

사양·설계·결과물을 사람이 이해해야 함

병목은 사람과 팀으로 옮겨 감

발표 자료의 질문: AI로 무엇을 할지에 더해, **AI를 쓰는 사람과 팀은 어떤 상태여야 하는가**

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07). 인용 출처(보충): quoteinvestigator.com/2016/06/26/shape/ — J. M. Culkin, Saturday Review, 1967

1.4 사람의 대기 시간이 전체 속도를 결정

KEY INSIGHT 사람이 끼는 의뢰·승인 단계가 병목이 됨 — 같은 기간에 돌릴 수 있는 개선 횟수가 크게 벌어짐

지금

사람 공정에 AI를 더함

기존 업무 흐름은 그대로 두고
일부 작업만 AI로 빠르게 처리

다음

사람의 대기 시간이 병목

AI 작업은 끝났는데 사람의 확인·
답변을 기다리는 시간이 속도를 정함

도달점

AI 만으로 끝나는 공정

의뢰·승인·인수인계 없이
AI가 목적까지 바로 진행

업무 한 건을 끝내는 시간 단위



막대 길이는 개념 비교용이며 비율 실측값이 아님

속도 차는 품질 개선과 시도 횟수의 차로 이어짐

사람에게 한 번 의뢰하는 동안 AI 와는 고치고 다시 해 보는 반복을 여러 번 돌릴 수 있음

Source: 원 발표 7~8쪽(Frieve-A, 2026.08.12). 시간 단위는 원 발표 기준

1.4 AI 코딩 도구 전원 배포 실증

KEY INSIGHT 구현이 빨라져도 개발 전체는 같은 배율로 빨라지지 않았음 — 다음 병목 위치를 먼저 찾아야 투자 대상을 정할 수 있음

실증 대상

Recruit HR Tech 부문 Indeed PLUS

개발 조직 엔지니어 전원

배포한 도구

2025-06 Claude Code

2025-09 Codex

관찰 방식

사용 방식은 지정하지 않음

현장마다 무슨 일이 생기는지 관찰

공정 흐름: 구현만 빨라지고 앞뒤 공정이 다음 병목 후보가 됨



AI 논의가 흔히 모이는 두 질문

① 일자리가 사라지는가

② 어느 모델이 더 똑똑한가

발표 자료가 던진 세 번째 질문

병목은 어디로 옮겨 가는가

구현이 빨라져도 개발 전체가 같은 배율로 빨라진 현장은 없었음

발표 자료 관찰 기준

Source: 원 발표 2~3쪽 (Indeed Recruit Technologies 사내 실증)

1.4 단계별 효율 개선의 누적 구조

KEY INSIGHT 가장 느린 단계가 전체 배수를 정함 — 코드 작성만 빨라지고 리뷰·테스트가 그대로면 체감 효율이 오르지 않음

단계별 개선 배수가 겹쳐 전체 효율 N이 됨

기획·설계 n1 × 조사·평가 n2 × 코드 설계 n3 × 코드 구현 n4 × 테스트 n5 = 전체 효율 N배

병목 단계가 배수를 깎음(보충)

- 코드 구현이 빨라져도 리뷰·배포 승인은 그대로인 경우가 많음
- 대기 시간이 남으면 전체 리드 타임은 거의 줄지 않음
- **가장 느린 단계가 체감 효율의 상한이 됨**

단계별로 따로 재야 보임(보충)

- N은 발표 자료의 개념 표현이며 측정값이 아님
- 단계별 작업 시간과 대기 시간을 나눠 기록
- **기록이 있어야 막힌 단계를 찾을 수 있음**

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 5쪽(업무 효율 N배 개념)

보충: 배수의 곱 구조와 병목 해석은 원 발표 도식을 풀어 설명한 보충이며 수치는 넣지 않음

1.4 AI 도구 도입만으로는 생산성이 오르지 않음

KEY INSIGHT AI가 낮춘 것은 구현(How) 비용임 — 인계와 승인 절차가 그대로면 줄어든 시간이 대기 시간으로 빠져나감

발표 자료의 논지

1. AI가 낮춘 비용

How, 곧 무엇을 어떻게 만드는 일의 비용

2. 그 결과

현장에서 배우는 루프를 훨씬 빠르게 돌릴 수 있음

3. 리크루트의 선택

제품을 만드는 방식, 곧 프로세스 자체를 재설계

보충 **DORA 2025 AI 지원 소프트웨어 개발 보고서**

약 90%

업무에 AI를 쓰는 응답자 비율

기술 실무자 약 5,000명 대상 조사

AI는 증폭기로 작동

성과가 좋은 조직의 강점과
어려움을 겪는 조직의 약점을
함께 키움

사용률만으로는 성과 차이를 설명하지 못함

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22; Google Cloud DORA, State of AI-assisted Software Development 2025

오른쪽 카드는 원 발표에 없는 보충 자료

SECTION 1.5

일자리와 경제 전망

제번스 역설·WEF 전망·신입 채용 축소

KEY INSIGHT 단위 비용이 내려가면 총수요가 늘어남 — AI로 개발 비용이 내려가도 일감이 줄어든다고 단정할 수 없음

약 1세기

증기기관이 공장 동력을 대체하는 데 걸린 기간

하루아침에 일자리가 사라지지 않았음

3배

1900년까지 영국 석탄 소비 증가 (발표 자료 기준)

효율이 좋아졌는데 소비는 늘어남

증가

기계화 뒤 노동자 수

제품 가격 하락으로 수요가 그 이상 늘어남

제번스 역설 (Jevons Paradox)

1865년 W. S. Jevons, The Coal Question

효율 향상	석탄 1단위로 더 많은 일
↓	
단위 비용 하락	같은 결과를 더 싸게
↓	
용도 확대	새 산업·새 기계에 적용
↓	
총수요 증가	소비량은 오히려 늘어남

AI로 개발 단위 비용이 내려갈 때도 같은 구조를 먼저 의심함

Source: Jevons, The Coal Question (1865); Atack·Bateman·Margo; Bessen (원 발표 5쪽 인용)

KEY INSIGHT 새 일자리는 기술에 맞춰 일을 다시 짠 쪽에 생김 — 전기를 들이기만 한 공장은 약 30년간 생산성이 오르지 않았음

자동화가 만드는 두 효과

Acemoglu & Restrepo (2019) 정리

대체 효과 (Displacement)

기존 작업을 기계·소프트웨어·설비가 대체

사람이 하던 작업이 줄어듦

복권 효과 (Reinstatement)

사람이 비교우위를 갖는 새 작업이 생김

줄어든 만큼 새 일에 인력이 다시 배치됨

고용의 질은 바뀌고 총량은 새 균형으로 이동

전기 도입 공장의 두 갈래

동력원만 교체

- 증기기관 → 모터
- 천장 축과 벨트가 남음
- 배치는 동력에 종속
- 생산성 약 30년 정체

작업을 다시 배치

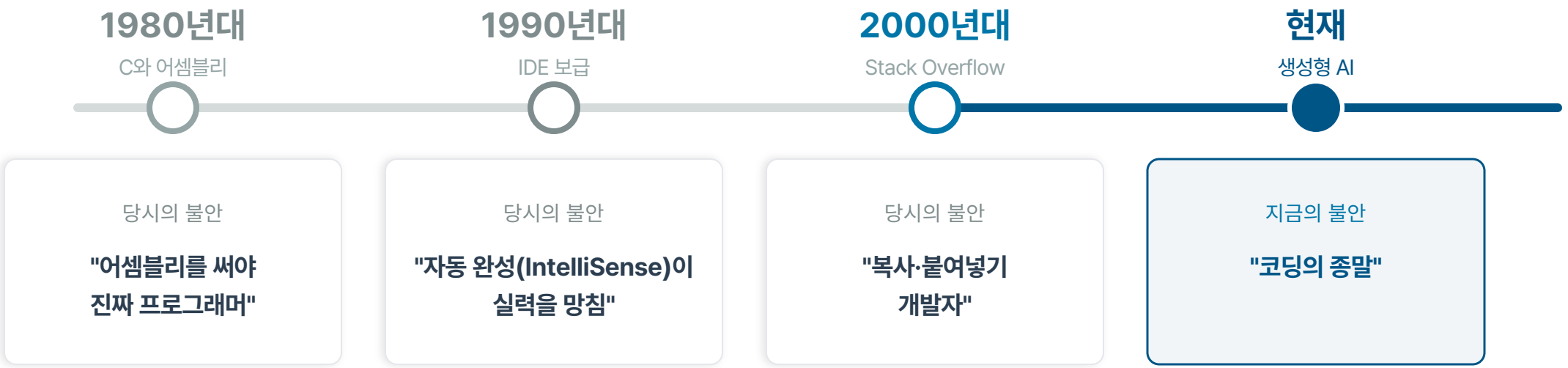
- 기계마다 소형 모터
- 작업 순서대로 재배치
- 공정이 동력에서 독립
- 효과가 여기서 나옴

복권은 저절로 오지 않음

기술에 맞춰 일을 다시 짠 쪽에 새 작업이 생김

Source: Acemoglu & Restrepo, "Automation and New Tasks," JEP 33(2), 2019; Paul David, "The Dynamo and the Computer," AER 1990 (원 발표 6~7쪽)

KEY INSIGHT 과거 전환기에도 엔지니어 직무는 사라지지 않았음 — AI도 직무 소멸보다 직무 이동으로 봐야 대응 방향이 보임



같은 흐름의 다른 예: 클라우드 등장 때의 "인프라 엔지니어 불필요론"

매번 직무가 사라질 것이라는 주장이 나왔지만, 실제로는 엔지니어가 다루는 추상화 수준이 한 단계씩 올라감

Source: 원 발표 p.7–8

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

KEY INSIGHT 일자리 총량보다 일자리 안의 업무 구성이 더 크게 바뀜 — 순증 7,800만 개와 함께 기존 기술의 39%가 바뀔 것으로 전망됨

1억 7,000만

새로 생기는 일자리

2025~2030년

9,200만

대체되는 일자리

정형 사무·입력 직군이 상위

39%

바뀌는 핵심 기술의 비중

현재 기술 가운데 2030년까지

조사	1위 역량	AI 관련 역량의 위치
WEF 고용주 1,000곳 이상	분석적 사고	수요가 가장 빠르게 느는 기술(AI·빅데이터)
Korn Ferry 채용 실무자 1,670여 명	비판적 사고(73%)	채용 평가 항목 5위

Source: World Economic Forum, The Future of Jobs Report 2025(2025.01) · Korn Ferry, TA Trends 2026(2025.10)

KEY INSIGHT 초급 개발자 채용은 줄고 AI 역량 요구는 늘어 양극화됨 — 일본과 같은 흐름이 국내에서도 확인됨

IT·통신 대기업·중견기업
신입 채용 공고

−73%

2026년 3월 공채 시즌, 전년 동월 대비

전 산업 대기업·중소기업
신입 채용 공고

−45%

1,438건 → 791건 (2026년 3월)

AI 역량 요구 공고
2026년 초 급증

기업의 AI 시스템 구축 확대로
AI 역량을 요구하는 공고 증가

일본 사례와 겹치는 지점

- AI 코딩 도구가 코드 작성부터 테스트까지 맡으면서 초급 개발자 채용이 줄었다는 보도
- 판교 IT 기업의 구조조정과 신입 채용 축소가 함께 진행
- 즉시 전력·사업 기여를 보는 채용 기준 변화는 국내 채용 시장 판단에도 참고가 됨

Source: 채용 플랫폼 캐치 집계 인용 보도(다음 뉴스, 2026) · ITWorld Korea "AI, 초급 인력 일자리 줄이며 노동시장 구조 재편 중" · 맨파워그룹 자료 인용 보도

보충: 원 발표에 없는 한국 자료 · 언론 보도 기준이며 고용노동부 공식 통계가 아님

KEY INSIGHT 고난도 업무가 빠지면 직무 수준이 낮아지고 정형 업무가 빠지면 높아짐 — 채용과 육성 방식도 직무별로 갈림

디스킬링: 직무 수준 하락

예: 여행사 직원, 테크니컬 라이터

AI가 가져가는 업무

복잡한 일정 계획(교육 13.5년 수준)

문장 수정 판단(교육 18.7년 수준)

사람에게 남는 업무

발권·수금 같은 관리·사무 업무

(교육 11.5~12.0년 수준)

기업 대응

사무 업무를 통합·자동화

신입 채용으로 단순 보충하지 않음

업스킬링: 직무 수준 상승

예: 부동산 관리자

AI가 가져가는 업무

일상적인 장부 정리와 임대료 검토

(교육 12.6~12.8년 수준)

사람에게 남는 업무

복잡한 협상과 이해관계자 관리

(암묵지·고급 역량)

기업 대응

정형 업무 감축

신입에게 협상 같은 고급 경험을 일찍 부여

Source: 원 자료 4쪽, Anthropic Economic Index 2026년 1월 보고서, The Decoder 보도(사례 해설)

직업 사례와 교육 연수는 원 자료 기준. 사례는 Anthropic 원문에서 직접 확인하지 못해 보도 해설로 대조함

SECTION 1.6

목적 중심 업무 지시

목적과 완료 기준을 적는 지시서

1.6 AI에 과업을 통째로 맡기는 비중

KEY INSIGHT 업무 시스템에 연동한 사용은 통째 위임이 64%로 대화형의 두 배 — 실행이 시스템 안으로 들어가면 사람은 무엇을 맡길지 정하는 역할로 이동함



대화형 측정	지시형	자동화	증강
2025년 1월	27%	41%	56%
2025년 8월	39%	증강보다 높음	—
2025년 11월	32%	45%	52%

개인 채팅 도구 배포에서 멈추면 위임 폭은 30%대에 머묾 — 업무 시스템 연동과 업무 재설계가 위임 범위를 넓힘

Source: Anthropic, Anthropic Economic Index report(2026.01, 2025년 11월 수집 자료)

1.6 방법 중심 지시서와 목적 중심 지시서

KEY INSIGHT 목적·대상·완료 기준·판정 근거를 적은 지시서는 그대로 AI의 작업 명세가 됨 — 방법만 적으면 결과를 판정할 기준이 없음

항목	방법 중심 지시서	목적 중심 지시서	점검 질문
목적	보도자료 작성	신제품 출시를 기존 고객에게 알려 업그레이드 문의를 늘림	이 일이 끝나면 무엇이 달라지는가
대상	명시 없음	기존 고객사 IT 운영 담당자	결과물을 누가 읽고 쓰는가
완료 기준	A4 2쪽, 금요일까지	핵심 변경 세 가지를 담당자가 이해할 수준으로 설명	무엇을 보면 끝났다고 판단하는가
판정 근거	팀장 확인	제품 사양서와 대조, 발행 후 고객 문의 수	누가 어떤 자료로 판정하는가

에이전트에 맡길 때는 고객 발송·자료 삭제·결제처럼 되돌리기 어려운 동작 앞에 사람 확인 지점을 함께 적음

Source: MSAP.ai, AI 시대 일하는 방식 백서 1.2절(2026.10) — 보도자료 예시

1.6 AI가 대체하는 설계·구현과 사람에게 남는 요구 정의

목적 중심 업무 지시

KEY INSIGHT 원하는 것을 말하는 주체는 바뀌지 않음 — 목적과 가치는 LLM이 대신 정할 수 없어 엔지니어 업무의 중심이 그쪽으로 옮겨 감

일반적인 소프트웨어 개발 공정

요구 정의
고객이 원하는 것과 제공할 가치

요건 정의
AI 산출물이 충족할 조건

설계
AI로 대체되는 중

구현
AI로 대체되는 중

사람의 일
무게가 이쪽으로 이동

AI의 일
사람은 결과를 리뷰

요구·요건 정의도 AI로 대체될까

일부에서 나오는 의견이지만 발표 자료는 어렵다고 봄

원하는 것을 말하는 주체는 사람
이렇게 하고 싶다고 요구하는 쪽은 언제나 사람

목적과 가치는 외부에서 주어지지 않음
AI가 대신 정할 수 있는 대상이 아님

결과
요구·요건 정의가 사람의 핵심 업무로 남음

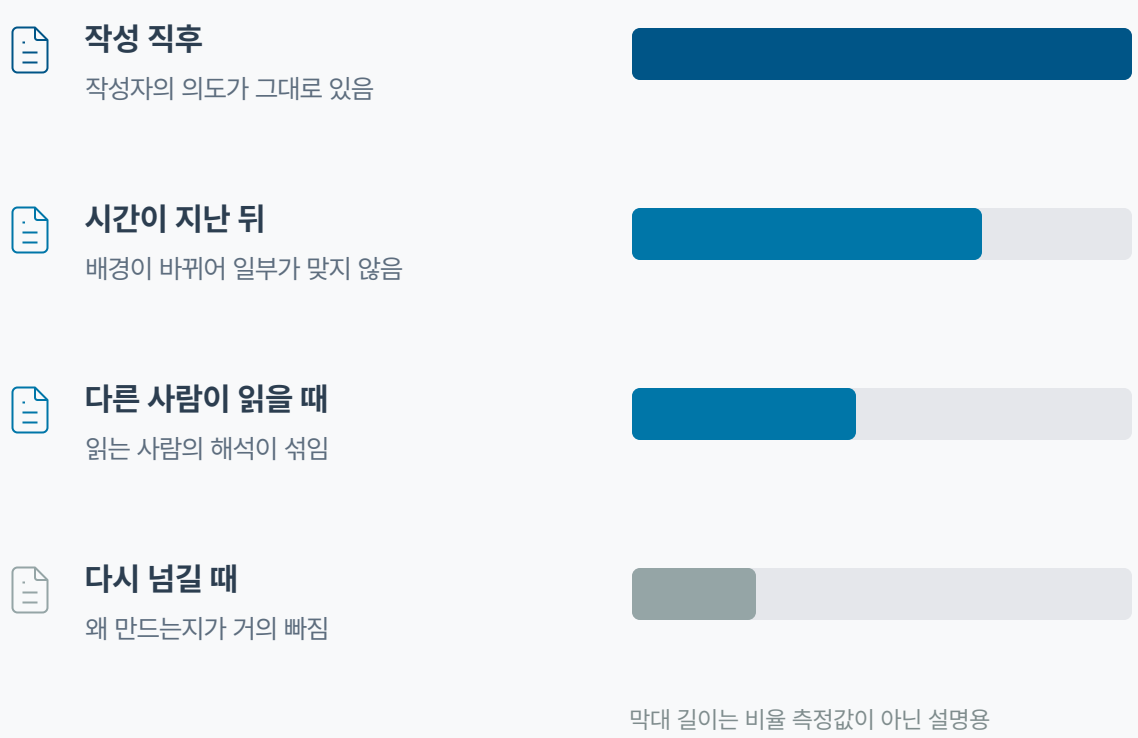
Source: 원 발표 자료(MinoDriven, 2026.06.17) 9~12쪽

공정 구분과 대체 전망은 원 발표자의 견해

1.6 Context는 작성 직후부터 낡기 시작함

KEY INSIGHT 문서를 자세히 쓰는 것만으로는 Why가 전달되지 않음 — 의도를 어떻게 전파할지가 Context 설계의 핵심임

문서에 남는 Why (개념도)



막대 길이는 비율 측정값이 아닌 설명용

발표 자료

Context 설계는 얼마나 자세히 쓰느냐만의 문제로 끝나지 않음

써 둔 순간부터 낡기 시작하는 Why를
어떻게 전파할지 설계하는 일이 핵심
사람을 거칠 때마다 악의 없이 Context가 빠짐

보충 **AI 쪽에서도 같은 문제가 있음**

Anthropic은 Context가 길어질수록 모델이
필요한 정보를 정확히 찾지 못하는 현상을
context rot으로 설명함

많이 넣기보다 필요한 것만 골라 넣는 설계가 필요

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 23·25쪽; Anthropic, Effective context engineering for AI agents (2025)

오른쪽 아래 카드는 원 발표에 없는 보충 자료

KEY INSIGHT 기획과 구현을 한 사람이 맡으면 PRD로 옮길 때 빠지던 의도가 그대로 남음 — 개인 역량 확장이 의도 손실을 줄이는 수단이 됨

해법 하나

Context를 넘기지 않음

PRD에 옮겨 쓰면 빠지는 의도도
직접 만드는 사람 안에서는
그대로 남음

기획과 구현이 한 사람 안에서 이어짐

사례: 엔지니어 출신 멤버의 기획 겸무

맡은 역할
기획을 하고 싶다는 본인 의사에 따라 기획 업무를 겸무로 맡김

초기
How부터 생각하는 습관 때문에 고전

전환
점차 Why부터 생각하게 됨

성장
엔지니어링 역량을 유지한 채 전 과정을 이끌며 팀 생산성을 높이는 리더가 됨

시사점

사람을 거칠 때마다 악의 없이 Context가 빠짐

기존 역할에 묶이지 않고 개인 역량 확장과 조직·역할 재설계를 함께 다룸

SECTION 1.7

AI 활용 성과 격차의 원인

같은 도구에서 성과가 갈리는 이유

1.7 AI 활용 효과와 설계 지식의 관계

KEY INSIGHT 경험 많은 쪽이 생성 결과의 구조 결함을 찾아 고침 — 도구를 함께 도입해도 팀 사이 격차는 줄지 않음

어느 쪽 엔지니어가 AI 기술을 더 잘 활용할 수 있는가

설계 지식과 경험이 많은 엔지니어

발표 자료의 답

- 생성 결과의 구조를 평가하고 받아들일지 판단
- 고칠 곳을 바로 찾아 직접 수정
- 필요한 질문을 좁혀서 AI에 요청

설계 지식과 경험이 적은 엔지니어

결과를 검증할 기준이 부족

- 그럴듯한 결과를 그대로 받아들임
- 모듈 경계 위반 같은 구조 문제를 놓침
- 질문과 수정을 반복하며 시간이 길어짐

설계 지식과 경험이 AI 활용 효과의 차이를 만듦

AI는 판단 기준을 대신 주지 않음 — 결과를 고를 기준은 엔지니어의 설계 지식에서 나옴

Source: 원 발표(masuda220, Forkwell Library #100, 2025.7.10) 질문 장

원 발표는 질문과 두 엔지니어 구분만 제시, 카드의 세부 항목과 하단 설명은 해설 보충

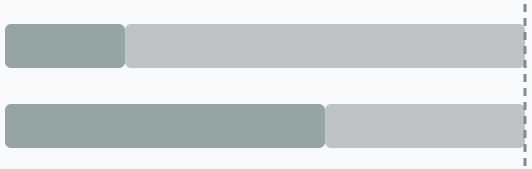
1.7 AI는 사용자의 능력을 증폭함

KEY INSIGHT AI의 도움을 받아도 자기 기술력을 넘는 결과물은 드묾 — 기반 역량이 결과 품질의 상한을 정함

AI 효과를 보는 두 가지 설

높임

누구나 비슷한 수준으로 끌어올림

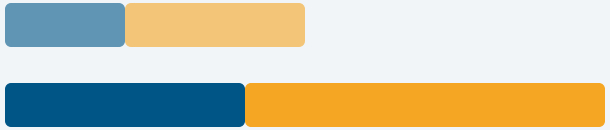


결과가 같은 선에 모임

발표 자료의 관찰과 맞지 않음

증폭

원래 능력에 비례해 결과가 커짐



기반 기술력 × AI = 결과

발표 자료는 이 설을 지지

기반 기술력 AI로 늘어난 결과(개념도, 측정값 아님)

자기 기술력을 넘는 결과물을 AI로 내는 사람은 드묾(발표 자료 관찰)

보충

DORA 2025 보고서

State of AI-assisted Software Development

약 5,000명

기술 실무자 설문 응답

핵심 결론: AI는 증폭기

잘하는 조직의 강점과
어려운 조직의 문제를 함께 키움

발표 자료는 개인 기술력, DORA는
조직 역량 기준으로 같은 현상을 봄

Source: Google Cloud DORA, 2025

Source: 원 발표 11쪽(uhyo, 2026.06.18) · DORA 2025 State of AI-assisted Software Development(dora.dev/research/ai)

케냐 창업자 640명 실험의 상·하위 격차

KEY INSIGHT 평균 효과는 0에 가까웠지만 상위는 +15%, 하위는 −8% — 격차는 질문 실력보다 AI에 맡긴 문제의 난이도에서 생김

+15%

상위 집단 매출·이익

기준선 성과가 중앙값보다 높은 창업자

−8%

하위 집단 매출

기준선 성과가 중앙값보다 낮은 창업자

≈0

전체 평균 효과

두 집단의 이득과 손실이 서로 지워짐

집단	질문 수·질문 품질	AI에게 맡긴 문제
상위 집단	차이 없음	비교적 단순한 과업에 도움을 구함
하위 집단	차이 없음	경쟁 심화·가뭄·생존 자금처럼 조언으로 풀기 어려운 문제를 물음

Source: Otis 외, UC Berkeley Haas·Harvard, 케냐 창업자 대상 무작위 대조 실험(2024)

1.7 토큰이 모자란 사람과 남는 사람

KEY INSIGHT 사용량 격차는 도구 숙련보다 맡길 목적의 양에서 생김 — 사용량은 신호로 두고 성과는 결과 지표로 잼

구분	토큰이 모자란 사람	토큰이 남는 사람
출발점	해결할 목적과 완료 기준이 먼저 있음	도구가 먼저 있고 쓸 곳을 찾음
위임 방식	여러 과업을 동시에 맡기고 결과를 검증	단발성 질문 뒤 결과 확인에서 끝남
필요한 지원	예산 상한 조정, 검증 기준 정비	위임할 업무 발굴, 목적 정의 코칭

사용량 순위를 목표로 삼은 Meta·Amazon 은 순위표 운영을 중단함 — 사용량은 위임 확산의 선행 신호로 두고, 예산과 평가는 완료 기준 충족률·손익 기여로 정함

Source: MSAP.ai, AI 시대 일하는 방식 백서 2.4절(2026.10) · IBM Think, Tokenmaxxing(2026)

1.7 AI를 학습에 쓰는 역량

KEY INSIGHT AI로 배우는 개발자와 AI에 기대 배우지 않는 개발자의 격차가 벌어짐 — 팀 교육 방식이 이 격차를 좌우함

학습 지원에는 강력함

모르는 개념을 바로 묻고 예제를 받으며
사람의 학습에 AI를 활용할 수 있음

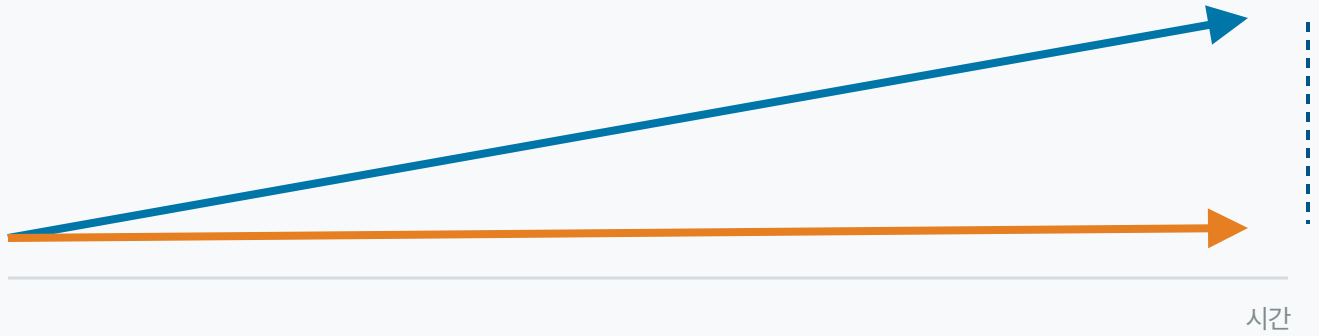
개념 설명 · 코드 해설 · 연습 문제 (보충 예시)

학습 동기는 떨어지기 쉬움

쉬운 일은 AI가 대신 해 주기 때문에
스스로 익힐 이유가 줄어듦

입문 단계 과제를 직접 풀 기회가 줄어듦

시간이 지날수록 벌어지는 격차



AI를 학습에 쓰는 사람
질문하고 확인하며 실력을 쌓음

AI에 기대 배우지 않는 사람
결과만 받아 쓰고 원리는 모름

Source: 기시다 나오키(LINE야후), GDG Kwansai 발표 자료, 2025-01-18. 격차 그래프는 원 발표 주장을 개념도로 그린 것이며 측정값이 아님

SECTION 02

직무별로 달라지는 일

사무직·마케팅·HR·영업·기획·개발·디자인·QA 직무별 변화

SECTION 2.1

경리·회계 사무의 변화

월 결산을 작업과 판단으로 나눠 본 경리 업무

2.1 WEF 감소 직군에 포함된 경리·기장 사무

KEY INSIGHT 감소 상위 직군은 대부분 정형 사무직임 — 입력·전기 위주의 경리 업무가 먼저 영향받는다는 뜻임

감소 속도 상위 7개 직군 (WEF, 2025~2030 전망)



막대 길이는 순위를 나타냄 (감소율 수치 아님)

경리 · 기장 · 급여 사무

약 20% 감소

2030년까지 전망 (발표 자료 기준, 미확인)

감소 직군의 공통점

입력 · 전기 · 대조 같은 정형 사무가 중심

경리도 같은 성격의 작업 비중이 큼

보충: 1,000여 개 기업 · 55개 경제권 대상 조사

Source: World Economic Forum, Future of Jobs Report 2025(2025-01) · 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

감소율 20%는 원 발표 인용 수치로 WEF 원문 표에서 직접 확인하지 못함

KEY INSIGHT 분개 입력·전기·정형 보고서는 AI로 대체됨 — 판단과 설계에 쓰는 지식·경험의 가치는 오히려 커짐

표층 작업

AI로 대체됨

- 분개 입력
- 전기 · 집계
- 정형 보고서 작성

절차가 정해져 반복되는 일

본질 업무

그대로 남음

- 증빙 수집 · 판단
- 데이터 정리 · 구조화
- 업무 흐름 설계 · 유지
- 세무 · 회계 처리 판단

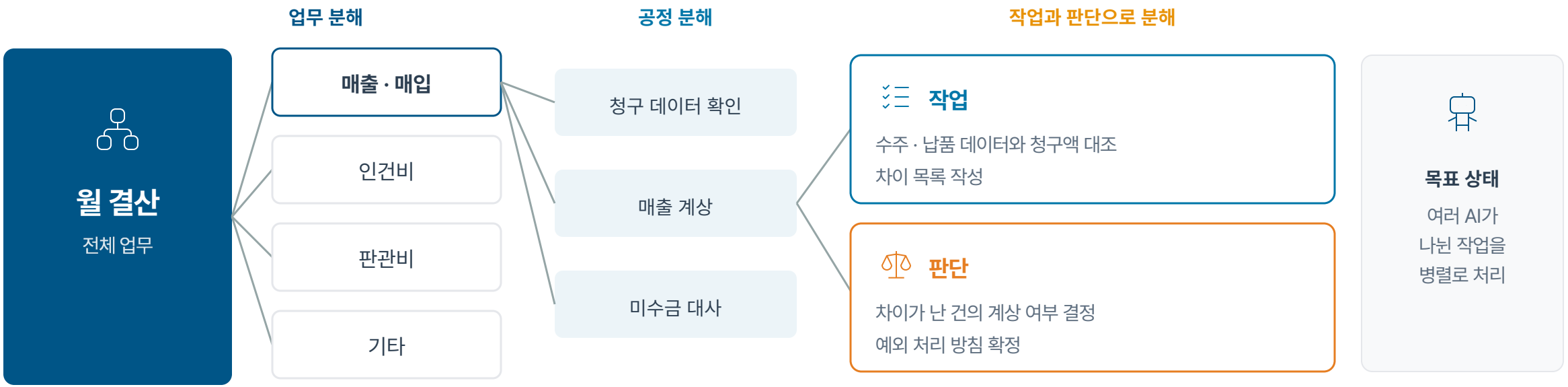
지식과 경험이 경쟁력이 되는 일

업무를 대하는 방식이 바뀜

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

KEY INSIGHT 한 줄 지시로는 월 결산이 처리되지 않음 — 업무 구조를 알고 적절한 크기로 나눌 수 있는 사람이 AI를 쓸 수 있음

"이번 달 결산 해 둬" — 이 한 줄로는 에이전트가 월 결산을 처리할 수 없음



공정 · 작업 · 판단 항목은 매출 · 매입 업무를 예로 든 설명용 예시 (보충)

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

KEY INSIGHT AI는 결정에 필요한 재료를 모으는 데까지만 도움 — 최종 결정과 설명 책임까지 넘기면 문제가 생겼을 때 책임질 사람이 없음

작업

절차가 정해져 있음

반복해서 처리할 수 있음

누가 맡는가

시스템과 AI가 지원

입력 · 전기 · 대조 · 초안 작성

판단

상황을 읽고 정함

결과에 책임을 짐

누가 맡는가

AI는 판단 재료 수집까지만 지원

최종 결정과 설명 책임은 사람이 맡음

AI가 결정까지 대신하면 문제가 생겼을 때 설명할 사람이 없음

판단 재료는 AI가 모으고, 결정과 설명은 담당자가 맡는 구조로 설계

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

KEY INSIGHT AI가 일부라도 맡을 수 있는 판단은 다섯 중 둘뿐임 — 나머지 셋은 경험과 경영 이해가 있어야 성립함

판단 유형	경리 업무 장면	AI가 맡을 수 있는 범위	AI 관여
평가 판단	분개 · 서류 · 처리의 적정성 확인	판단 재료 제시만 가능	△
맥락 판단	사내 규정 · 과거 관행과 개별 건 대조	선택지 제시만 가능 예외 판단은 사람	△
트레이드오프 판단	마감 속도와 정확성, 비용 절감과 업무 품질	선택지 정리만 가능 결정은 사람	×
예측 판단	결산 전망 · 현금흐름 영향 예측	경험과 경영 이해 없이 성립하지 않음	×
당사자성 판단	의견이 갈리는 처리의 방침을 미루지 않고 결정	대체 불가	×

△ 일부 지원 가능 × 사람이 맡음

Source: David S. Duncan, Harvard Business Review(2026)의 판단 분류를 원 발표가 경리 업무에 대입 · 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

2.1 작업은 AI가 맡고 회계 판단은 담당자가 맡음

KEY INSIGHT 서류 정리·입력·검색·초안은 맡길 수 있음 — 단 처리 규칙과 데이터가 먼저 정리돼 있어야 함

AI가 맡을 일

업무와 데이터가 정리돼 있을 때

- 서류 수령 · 정리 (데이터화)
- 정형 입력과 전기
- 검색 · 대조 · 요약
- 문서 초안 작성

경리만 할 수 있는 일

판단과 신뢰가 필요한 일

- 거래 실태에 근거한 회계 처리 판단
- 결산 수치의 정확성 · 신뢰성 확보
- 경영진 대상 재무 보고와 제언
- 부서 간 신뢰 관계 구축

작업에서 벗어난 시간만큼 경리 본연의 판단 · 보고 업무에 집중할 수 있음

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

SECTION 2.2

사무직의 업무 설계 역할

업무 정의 시트와 업무 흐름도로 일하는 사무직

KEY INSIGHT 평가 기준이 자리를 채우는 일에서 매출·이익 기여로 옮겨 감 — 작업만 하는 자리는 AI와 인력 감축으로 줄어듦

01

평가 기준 이동
머릿수 제공 → 사업 기여

"사람이 있음"보다
매출·이익에 얼마나 기여했는지를
보고 평가함

02

순수 작업자의
절대 수 감소

AI 도입 · 인력 감축 · 연령 조정이
함께 진행되며
작업만 하는 자리가 축소됨

03

돈을 내는 사람의
요구 관찰

고객·경영진이 무엇에
기꺼이 돈을 내는지
파악하는 능력이 중요해짐

아무것도 하지 않고 정년까지 갈 수 있는지 점검이 필요한 시점

먼저 현재 직장에서 할 수 있는 일부터 검토

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07)

2.2 AI 활용의 전제인 업무 구조화

KEY INSIGHT AI는 정리되지 않은 업무 위에서 제대로 동작하지 않음 — 사람의 역할은 실행보다 업무 설계 쪽으로 옮겨 감

질문 AI가 있으니 업무 설계는 필요 없다는 주장은 맞는가 — 도서 요약은 세 가지 이유로 반대 결론

01

업무 구조 명확화

AI는 정리되지 않은 업무
위에서 제대로 동작하지 않음

로봇 청소기를 돌리기 전에
바닥을 먼저 치워야 하는 것과 같음

02

프롬프트와 업무 이해

좋은 프롬프트를 쓰려면 업무의
목적과 구조를 깊이 알아야 함

무엇을 입력으로 주고 어떤 결과를
받을지 정하는 일이 곧 업무 설계

03

거버넌스와 설명 책임

AI 판단 근거를 설명할 책임은
사람에게 남음

설명하려면 어느 단계에서 무엇을
판단하는지 프로세스가 분명해야 함

결론 사람에게 요구되는 역량은 업무를 구조로 파악하고 설계하고 계속 고치는 업무 설계 역량

Source: '업무 설계 교과서'(Shunsuke Takeuchi, Gijutsu-Hyoron, 2025) NotebookLM 요약 슬라이드 11쪽

세 이유와 결론은 도서 요약 기준. 02·03 카드 아래 줄의 풀이는 보충

2.2 업무 설계의 정의와 세 가지 작업

KEY INSIGHT AI 도입보다 업무 가시화와 정리가 우선임 — 정리되지 않은 업무는 AI에 넘길 단위가 없음

업무 설계란

담당자 · 처리 대상 · 처리 순서 · 처리 수단 — 네 가지를 정하는 일

업무 흐름 가시화

부서를 넘는 처리 순서를
한 장에 그림

작업과 판단 구분

반복 가능한 일과
책임지고 정할 일을 나눔

사람과 AI의 역할 결정

AI에 넘길 작업과
사람이 맡을 판단을 정함

실무 도구 두 가지: 업무 정의 시트(업무를 말로 정리) · 업무 지도(공정을 한 장에 시각화)

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

2.2 업무 정의 시트의 다섯 요소

KEY INSIGHT 흐름도보다 먼저 업무를 말로 정리함 — OUTPUT과 GOAL이 서로 맞는지가 가장 자주 어긋나는 지점임

SCOPE 대상 범위	업무가 다루는 범위의 정의
INPUT 입력	처리에 필요한 정보와 서류
DO 처리	실제로 하고 있는 처리 내용
OUTPUT 출력	**완료로 보는 상태와 그 산출물**
GOAL 목적	**무엇을 달성하려고 하는 업무인지**

서로 맞는가?

작성 원칙

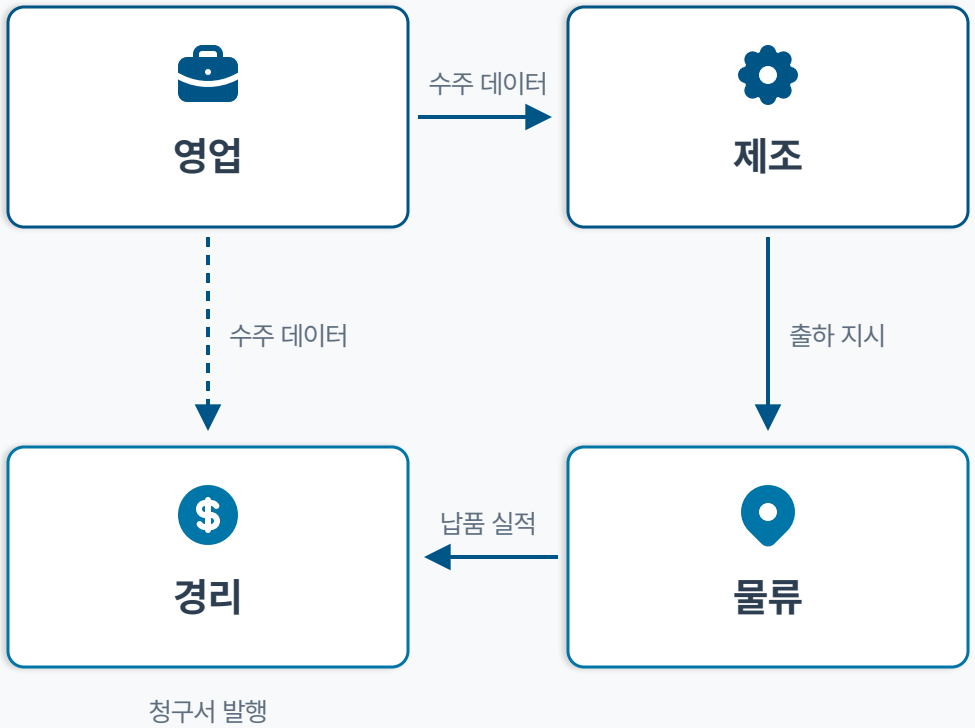
- 업무 전체를 **A4 한 장에 말로 정리**
- 흐름도보다 **먼저 작성**
- 현재 상태를 **그대로 적음**
- 완벽하지 않아도 되는 **논의용 초안**

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

2.2 업무 흐름도 한 장으로 전체 공유

KEY INSIGHT 부서마다 다르던 업무 인식이 한 장의 그림으로 맞춰짐 — 이후 개선 논의의 공통 기준이 됨

업무 흐름도 예시 — 부서 간 수주부터 청구까지



업무 흐름도가 필요한 이유

전체 공유

부서·담당자마다 다른 업무 인식을 맞춰
한 팀으로 움직이는 바탕 마련

속인화 방지

특정 담당자 머릿속의 암묵지를 누구나 아는
형식지로 바꿔 업무 연속성 확보

과제 발견

개별 작업만 봐서는 안 보이던 병목과
불필요한 재작업 확인

Source: '업무 설계 교과서'(Shunsuke Takeuchi, Gijutsu-Hyoron, 2025) NotebookLM 요약 슬라이드 5쪽

도서의 '업무 지도'를 업무 흐름도로 표기. 출하 지시·납품 실적 화살표 이름은 보충

KEY INSIGHT 증빙과 분개 뒤에 있는 업무 흐름에 관심을 갖는 것이 첫 단계임 — 기존 회계 지식에 설계 능력이 더해질 때 AI를 수단으로 부릴 수 있음

1 AI가 발전해도 경리의 본질은 그대로

증빙 판단 · 업무 흐름 설계 · 세무 판단은 사람의 몫

2 경리가 출발점이 되는 업무 재설계가 중요

현장(상류)부터 데이터와 프로세스를 정리

경리 담당자가 오늘 시작할 일

업무에 관심 갖기

증빙과 분개 뒤에 있는
업무 흐름을 살펴봄

시트 한 장 써 보기

담당 업무 하나를
업무 정의 시트로 적음

설계를 먼저 하기

시스템과 AI는 수단으로 보고
업무 설계부터 시작

기존 지식과 경험에 업무 설계 능력을 더하면 작업자에서 업무 설계자로 옮겨 갈 수 있음

Source: 다케우치 슌스케, Money Forward CONNECT with AI 발표(2026-05-26)

SECTION 2.3

마케팅·HR 직무의 변화

콘텐츠 제작과 채용 업무에서 사라지는 일과 남는 일

KEY INSIGHT 반복 제작은 에이전트로 넘어가고 기획·검수·성과 판단이 마케터에게 남음 — 독자와 금칙어를 아는 마케터가 제작 에이전트를 설계함

사라지는 일

반복 제작

- 초안 작성과 문안 변형의 반복
- 배너·브로셔 같은 정형 디자인
- 채널별 형식 변환과 재가공

남는 일

판단과 책임

- 캠페인 목적과 대상 독자 정의
- 사실·브랜드 검수
- 성과 판단과 다음 기획

새로 생기는 일

에이전트 설계·운영

- 콘텐츠 제작 에이전트 설계·운영
- 검수 기준(금칙어, 사실 확인 규칙) 정비
- 성과 데이터를 에이전트 개선에 반영

평가 지표도 발행량에서 리드·전환 기여 같은 결과 지표로 옮겨야 함

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.2.1절(2026.10)

KEY INSIGHT 블로그는 주 1편에서 하루 5편, 백서는 1주에서 생성 5분·검토 1시간 — 편수가 차이를 만들지 못하니 보고 항목을 반응 지표로 바꿈

주 1편 → 하루 5편

블로그 발행량

1주 → 5분 + 1시간

백서 한 편

생성 5분, 사람 검토 1시간

구독 해지

디자인 SaaS

브로셔 제작을 에이전트 스킬로 옮김

보고 항목	과정 보고(전)	결과 보고(후)
산출	이번 달 블로그 60편 생성	블로그 경유 문의 건수와 영업 기회 전환 건수
원칙	만든 편수가 성과	만드는 건 의미 없음, 리뷰와 반응에 집중

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.2.1·7.1.2절(2026.10) — 국내 한 IT 기업 사례

KEY INSIGHT 채용 리더의 52%가 2026년에 자율 에이전트를 팀에 추가할 계획 — 선별은 에이전트가 맡아도 기준 설계와 편향 점검 책임은 HR에 남음

52%

자율 에이전트 도입 계획

인재 채용 리더 응답, 2026년 기준

사라지는 일	남는 일	새로 생기는 일
이력서 1차 선별과 면접 일정 조율	최종 채용 판단과 후보자 경험 관리	채용 에이전트의 선별 기준 설계와 편향 점검
직무기술서 초안 작성	직무 요건 정의	사람과 에이전트가 함께 일하는 팀의 직무 설계
정형 문의 응대	평가 체계 운영	문제 정의·검증 역량을 재는 평가 과제 개발

Source: Korn Ferry, Top Talent Acquisition Trends Shaping 2026(2025.10) · MSAP.ai 백서 6.2.2절

2.3 채용 1차 면접 앞에 AI 사전 면접 배치

KEY INSIGHT 지원 동기 확인과 회사 소개를 AI 사전 면접이 맡으면 면접관은 업무 역량 확인에 집중함 — 지원자 쪽에서도 정보 부족을 줄이는 수요가 확인됨

기존 채용 절차

지원 → 일정 조율 → **1차 면접** → 2차~최종 면접

지원 동기 확인·회사 소개에 30~60분으로는 부족

AI와 협업하는 채용 절차

지원 → **AI 사전 면접** → 일정 조율 → **1차~최종 면접**

차이가 있으면 지원자가 사전 질문 가능

업무 역량 확인에 집중

발표 자료 기업이 이 업무를 고른 이유

- 채용은 성장 중인 회사에 꼭 필요한 업무이고 사람이 조직의 근간
- 모든 지원자와 면접하기는 시간상 현실적이지 않음
- 1차 면접은 정보를 묻고 회사를 소개하는 정형 대화가 많음
- 담당자 시간은 한정적이라 사전 확인과 조율을 AI가 대신함

발표 자료 기업의 자사 서비스 사례임

이직 경험자 대상 설문 (n=400)

이직할 때 기업 정보 부족을 느낌



85%

특별히 없음 15%

사전 면접이 있으면 이용하고 싶음



78%

적극 이용 13% + 약간 이용 65% · 보통 17% · 비희망 6%

조사 단서

발표 자료 기업의 자체 온라인 조사

2024-05, 이직 경험자 400명 (일본)

외부 검증 자료 없음 — 참고 수치로 볼 것

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 22~24쪽(자체 온라인 조사, 2024-05, n=400)

KEY INSIGHT 조달할 수 있는 자원의 범위가 사람 밖으로 넓어짐 — 인사 부서가 AI도 일하기 쉬운 조직을 설계해야 함

새로 열린 영역 · AI

예전에는 조달할 수 없던 자원

조달 가능 영역 · 채용 후보자

지금까지 인사가 고려하던 범위

실제로 다루는 영역 · 직원

사람과 일한다는 전제 안의 범위

무의식적 제약

사람과 일한다는 전제가 조달 범위를
직원과 후보자로 한정함

외부 환경 변화

조달할 수 없던 자원을 AI로 쓸 수 있게 됨
발표 자료는 쇄국이 풀린 상황에 견줌

인사의 과제

AI를 조달 자원으로 보고 조직을 최적화함
발표 자료자는 이 인식이 아직 낮다고 봄

Source: Ryosuke Yoshizaki(Wadan Inc.), AI 시대의 조직 설계론, HR Tech Vision 2025 발표 자료
원 발표자는 사람을 자원으로 부르는 데 거부감이 있으나 설명을 위해 이 표현을 썼다고 밝힘

SECTION 2.4

영업 직무의 변화

B2B 구매자의 변화와 영업 평가 기준

2.4 영업직 중점이 효율에서 관계 구축으로 이동

KEY INSIGHT SaaS 붐이 끝나고 대기업·비IT 고객이 주력이 됨 — 온라인 상담 횟수보다 장기 관계가 매출을 좌우함

1

코로나 시기

효율 · 규모 중심

· 유동성이 풍부해 SaaS 붐

· SMB·스타트업 대상 박리다매

· The Model 방식의 영업 분업
IS · FS · CS 단계별 표준화

· 온라인 상담 횟수 확대

2

2023년

스타트업 버블 종료

· 투자 단계에서 회수 단계로 전환

· 판매처인 스타트업도 축소

· 대기업용 SaaS 커스터마이징

· 주력 고객을 엔터프라이즈·
비IT 기업으로 전환

3

AI 확산 · SaaS 회의론

관계 구축 중심

· SaaS가 주류에서 밀려남

· 사업회사가 수탁 개발과
무형 상품 납품에 주목

· **기존 분업형 영업 방식이
매출의 병목이 됨**

고객과 오래 관계를 쌓는
영업 인력의 가치 상승

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07)

The Model: 마케팅·IS(Inside Sales)·FS(Field Sales)·CS(Customer Success)로 영업 단계를 나누는 분업 모델

2.4 기업이 찾는 영업은 결재권자 밀착형

KEY INSIGHT 경쟁 입찰 전에 고객 내부로 들어가는 영업이 부족함 — 효율화 10년 동안 이런 인력이 영업 조직에서 빠졌기 때문임

기업이 원하는 영업 방식

결재권자 가까이에 머무는 영업

필요할 때만 찾아오는 영업에는 결재권자가 승인하지 않음
실무 담당자만 모은 설명회는 "공부가 됐다"로 끝남

경쟁 입찰 전에 고객 내부로 진입

가벼운 PMO 건으로 들어가 담당 영역을 넓히는 방식
신흥 컨설팅사가 주로 쓰는 진입 경로

"컨설팅 출신 선호"의 실제 의미

원하는 것은 직함보다 끈질기게 매출을 만드는 사람
효율화 과정에서 "고객과 어울리는 영업 인력"이 사라짐

부족한 이유

신입 인재가 컨설팅사로 몰리면서
벽에 부딪힐 때까지 밀어붙이는
젊은 영업 인력이 줄어듦

"왜 해야 하나요?", "비효율적이지 않나요?"
— 움직이기 전에 안 할 이유부터 찾는 경향

체육계 경험 재평가

약 150 개사

2026년 졸업생 채용 조건에 체육계 경험을
명시한 기업 수 — 5년 사이 3배 이상

자기소개서 빈출어: 체육계 "끈기" · 일반 학생 "효율화"

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07) — 원 발표가 인용한 니혼게이자이신문(2026) 보도와 대기업 관리자 인터뷰

2.4 B2B 구매자의 AI 사용과 영업 수요

KEY INSIGHT 구매자의 67%는 담당자 없는 구매를 선호하지만 69%는 AI 분석을 담당자에게 확인받기를 원함 — 영업 수요가 정보 전달에서 판단 검증으로 옮겨 감

67%

담당자 없는 구매 선호

2026년 3월 발표, 구매자 646명

45%

최근 구매에 AI 사용

같은 조사

69%

AI 분석을 담당자에게 확인받고 싶음

2026년 5월 발표

구매 단계	구매자가 하는 일
정보 탐색	제품 기능·가격 정보는 AI와 웹으로 스스로 찾음
판단 확인	AI로 내린 결론이 맞는지 영업 담당자에게 확인받음

Source: Gartner, B2B 구매자 조사(2026.03, 646명) · Gartner, AI 분석 검증 조사(2026.05)

2.4 영업 직무에서 사라지는 일과 남는 일

KEY INSIGHT 정보 전달과 문서 초안은 AI로 넘어가고 문제 진단·협상·관계가 남음 — 고객이 AI로 얻은 분석을 검증하는 일이 새로 생김

사라지는 일

정보 전달

- 제품 기능·가격 정보 전달
- 제안서·견적서 초안 작성
- 반복 후속 연락과 일정 관리

남는 일

진단과 협상

- 고객 문제 진단과 관계 관리
- 협상과 계약 조건 판단
- 내부 기술·제품 조직과의 조정

새로 생기는 일

판단 지원

- 고객이 AI로 얻은 분석의 검증·보정
- 고객 판단을 돕는 영업 지원 에이전트 제작
- 이해관계자별 판단 근거 정리

사례: 한 IT 기업의 기술영업(SA) 담당자는 고객 판단을 돕는 에이전트를 개발 부서에 맡기지 않고 직접 만들어 씀 — 고객이 묻는 요건과 결정 근거를 가장 잘 아는 사람이기 때문

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.3절(2026.10)

2.4 영업 평가 지표의 전환안

KEY INSIGHT 방문·통화 횟수는 정보 전달 시대의 활동량 지표임 — 고객 판단을 얼마나 도왔는지를 재는 지표로 바꿔야 역할 전환이 정착함

평가 영역	기존 지표	전환안
활동량	방문·통화·메일 횟수	구매 단계별 판단 지원 건수
정보 제공	자료 발송 건수	고객 질문에 검증된 근거로 답한 비율
성과	수주 금액	수주 금액과 판단 지원 뒤 다음 구매 단계로 넘어간 비율
역량	제품 지식	제품 지식과 영업 지원 에이전트 제작·운영

지표만 바꿔서는 정착하지 않음 — 도메인을 아는 영업 인력이 에이전트를 직접 만들 수 있는 환경을 함께 마련

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.3.2절(2026.10)

2.4 시장에서 선택받는 전문성의 조건

KEY INSIGHT 선택지가 좁고 믿을 만한 전문성이 보이는 쪽이 선택됨 — 생성형 AI는 할 일을 넓히기보다 한 영역을 깊이는 데 쓸 때 효과가 큼

현상

생성형 AI로 할 수 있는 일이 늘어남

제공자가 내세울 수 있는 항목이 많아짐

↓

결과

고객의 의사결정이 어려워짐

무엇이 강점인지 판별하기 어려워 선택을 미룸

↓

선택 기준

좁은 선택지와 믿을 수 있는 전문성

맡길 영역이 분명한 쪽이 먼저 선택됨

생성형 AI의 좋은 사용법

권하지 않는 쪽	권하는 쪽
범위 확장	**전문성 심화**
할 수 있는 일 늘리기	한 영역 더 깊이 파기

발표 자료 결론: 할 수 있는 일을 넓히기보다
깊이는 쪽이 생성형 AI의 좋은 사용법임

예: 같은 도메인의 요구 분석·검증·운영 자동화를
AI로 빠르게 반복해 실적 수와 품질을 함께 높임 (보충)

Source: 원 발표(AWS 재팬, 2025) 시장에서 배울 수 있는 것 쪽

왼쪽 흐름의 설명 줄과 오른쪽 아래 예시는 보충

SECTION 2.5

기획·PM 직무의 변화

PM 이 맡는 판단·합의와 제품용 프롬프트

2.5 PM 실력은 판단을 성립시키는 일

KEY INSIGHT 목표·판단 기준·합의 중 하나라도 빠지면 결정이 내려져도 팀이 따르지 않음 — 세 가지가 함께 있어야 판단이 실행으로 이어짐

목표 그리기

무엇을 위해 하는가

프로젝트가 끝났을 때
무엇이 달라져야 하는지 정함

성공 기준이 없으면
AI 산출물을 맞춰 볼 곳이 없음

판단 기준 세우기

무엇을 버리는가

범위·품질·일정이 부딪칠 때
무엇을 먼저 지킬지 정함

버릴 것을 정해야
남길 것의 우선순위가 섬

합의 이루기

누가 무엇을 하는가

결정 내용과 역할을
이해관계자와 맞춤

합의가 없으면 결정이
실행으로 이어지지 않음

발표 요지

PM의 실력은 목표를 그리고 판단을 성립시키는 일

Source: 원 발표(i3DESIGN, AI 구동 개발 스터디 제7회) PM 실력 정의 쪽

세 항목과 질문은 원 발표 내용, 카드 아래 풀이는 보충

KEY INSIGHT 선택지 생성은 AI에 넘겨도 선택의 책임은 넘길 수 없음 — 이 경계를 정하지 않으면 AI 초안이 그대로 결정이 됨

업무	AI에 맡기기 좋은 일 (선택지·초안)	PM에게 남는 일 (선택·책임)
요구사항	인터뷰 기록 요약, 요구사항 초안 작성	**목표와 성공 기준 결정**
범위·우선순위	기능 후보 목록과 공수 추정안 작성	**범위에서 뺄 항목과 우선순위 결정**
일정·위험	일정안·위험 목록 초안 작성	**위험 수용 여부와 대응 책임자 결정**
이해관계자	회의록 요약, 보고 자료 초안 작성	**합의 도출과 역할 배분**
산출물 검토	형식·누락 항목 1차 점검	**채택·폐기 판단과 승인**

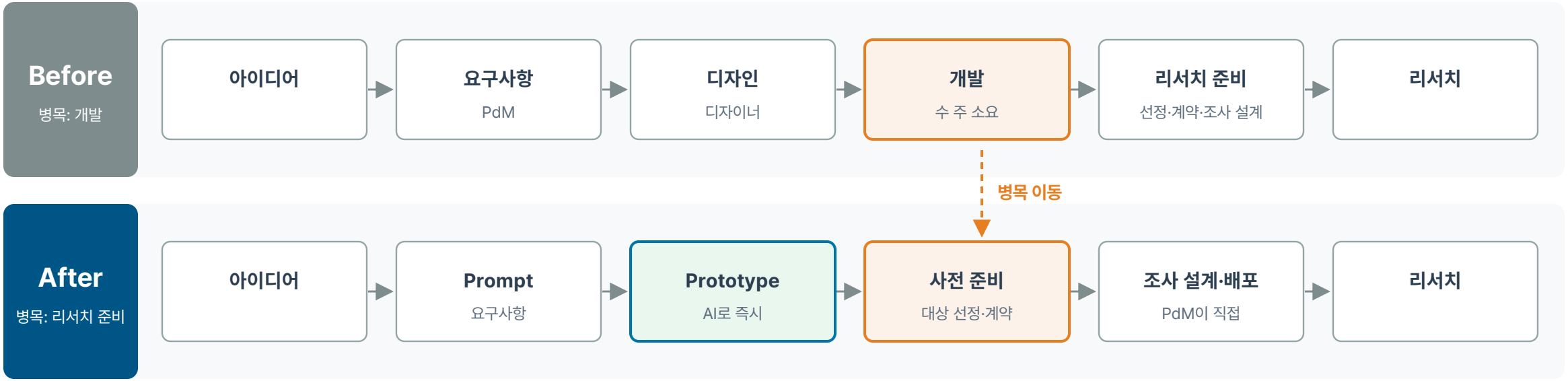
경계를 정하는 기준

되돌리기 어려운 결정, 사람·비용·계약에 영향을 주는 결정은 AI 초안을 참고하되 PM이 최종 판단

Source: 원 발표의 PM 실력 정의(목표·판단 기준·합의)를 업무별로 풀어 쓴 보충

이 장의 업무별 구분은 원 발표에 없는 예시

KEY INSIGHT 병목은 사라지지 않고 다음 단계로 옮겨감 — 조직이 프로세스를 계속 고쳐야 비연속적인 개선이 나옴



바꾼 리서치 절차

PdM → 책임자 승인 → 조사 설계·배포 → 실시

PdM이 책임자 승인을 받아 조사 설계와 배포까지 직접 맡음

리서치 준비 기간

수 주 → 최단 다음 영업일

발표 자료가 예시로 든 수치

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 20~21쪽

2.5 설계·구현이 빨라지자 기획이 병목이 됨

KEY INSIGHT 개발 공정만 자동화하면 전체 리드타임은 기획·검증 속도에 묶임 — 기획 프로세스도 AI 적용 대상에 넣어야 효과가 남



드러난 현상

· Software Factory가 빨라질수록 그 앞뒤의 기획과 검증이 전체 속도를 결정

· Factory 입구에 있는 기획 단계에서 안건이 쌓이기 시작함

freee의 대응

· AI로 기획 프로세스를 빠르게 하는 초고속 기획 프로젝트 시작

· 기획 문서 작성 시간을 줄이는 실험을 여러 제품에서 진행 (2026년 7월~)

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 16~17쪽

2.5 AI 이후 개발 흐름: PRD 단계 목업

KEY INSIGHT 동작하는 목업이 기획 단계에 나와 사업 부서 설명이 쉬워짐 — 엔지니어 관점이 들어가면 실제 개발의 기반으로 이어짐

Before · AI 이전

PRD → 디자인 → 설계 → 개발

- 설명은 문서와 이미지에 의존
- 동작하는 결과물은 마지막에 보임

기획·디자인 단계에서는 실제로 움직이는 화면 없이 합의를 진행함

After · AI 이후

PRD + 동작하는 목업
바이브 코딩으로 기획 단계에서 제작

→ **사업 부서 설명**
실물로 보여 주며 합의가 쉬워짐

→ **개발의 기반**
엔지니어 관점이 있으면 실물에 가까움

목업에 엔지니어 관점이 빠지면 보여 주기용 화면에 그치고, 들어가면 실제 개발의 출발점이 됨

Source: 원 발표(TRAILBLAZER Kazuhisa Naoi, 2026.08.21) 개발 흐름 변화 쪽

바이브 코딩: 자연어 지시로 AI가 코드를 만들게 하는 개발 방식. 하단 해설 문장은 보충

PM 업무가 AI 활용에서 AI 탑재로 이동함

KEY INSIGHT 도구로 쓰는 경험만으로는 제품 기능 판단이 서지 않음 — 기능을 기획하는 쪽은 LLM 동작을 따로 익혀야 함

지금

업무 도구로 AI 활용

발표 자료: AI를 쓰는 PM이 이미 많음

- 검색·잡무 자동화: ChatGPT, Gemini
- 경쟁 조사: Deep Research
- 디자인 목업: v0, Figma Make
- 회의록 작성: Notion, Gemini
- 코드 이해·작은 수정: Cursor, Claude Code
- 사내 업무 자동화: Gems, Dify, n8n

최근 1년 사이 LLM 발전과 함께 도구가 늘고 일하는 방식이 바뀜

다음 과제

제품 안에 AI 탑재

발표 자료: AI·에이전트 서비스는 계속 늘어남

- AI 브라우저·범용 에이전트·특화 에이전트
- 에이전트 기반 제품의 발전
- 기존 제품에 LLM 기능을 넣는 경우

발표 자료 관찰: 도구 활용에 비해 제품 탑재 사례는
아직 적음, PM은 AI를 제품에 넣는 쪽으로 이동

Source: Kenta Watanabe (LayerX), pmconf 2025 발표 자료 (2025.12.04)

도구 목록과 관찰은 원 발표 기준(2025.12). 제품명은 원 발표 표기 그대로

SECTION 2.6

개발자 역할의 이동

코드 작성에서 무엇을 왜 만들지로 옮겨 간 가치

KEY INSIGHT 어떻게 만들지의 많은 부분을 AI가 맡아도 설명 책임은 사람에게 남음 — 사업에 깊이 관여하는 엔지니어가 경쟁력을 가짐

구현 장벽이 낮아질수록

How

어떻게 만들지

- 많은 부분을 AI가 처리
- 구현 속도의 차이가 줄어듦

가치 이동

가치가 옮겨 가는 자리

What · Why

무엇을, 왜 만들지

- 만들 대상과 이유를 정하는 일
- 설명 책임은 이전보다 더 사람에게 남음

사업에 깊이 관여하는 엔지니어가 경쟁력을 가짐

무엇을 왜 만드는지 설명할 수 있는 사람이 AI가 만든 결과의 책임자가 됨

Source: 원 발표(TRAILBLAZER Kazuhisa Naoi, 2026.08.21) 역할 변화 쪽

원 발표는 사업 관여가 강한 사람이 살아남는다고 표현함. 구현 속도 차이 줄과 하단 둘째 줄은 보충

KEY INSIGHT How는 크게 바뀌어도 문제를 분석해 최적의 해법을 내는 Why·What은 그대로임 — 수단은 시기마다 다시 익혀야 함

How 크게 바뀜

- 자동 처리 구현에서 AI 적용까지 수단이 다양해짐
- 효율적인 수단이 시기마다 달라짐

Why · What 그대로 유지

- 문제 영역을 분석함
- 최적의 해법을 제공함

엔지니어의 역할

역할 1
가치 제공에 따르는 우연적 복잡성을 숨김

사용자와 사업 담당자는 기술의 번거로움을 보지 않고 가치만 받음

역할 2
시기에 맞는 효율적 수단을 계속 학습

자동 처리 구현이든 AI 적용이든 그때 가장 나은 수단을 고름

Source: 원 발표 27쪽·34쪽(오다카 도시유키, Kinoko Conference 2025)

역할 카드의 하단 설명 줄은 해설 보충

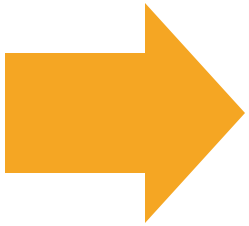
KEY INSIGHT 성과를 재는 기준도 바뀌어야 함 — 작성한 코드량보다 고객에게 전달된 가치가 기준

예전 기준

코드 작성

엔지니어의 시간 대부분이
코드를 직접 쓰는 데 들어감

성과를 작성한 기능·코드로 봄



지금 기준

가치 전달

기획·구현·출시·안내를 이어서 담당
목표는 고객에게 가치가 실제로 전달됨

혼자 다 하지는 않음 — 사내 여러 사람과 협력

남는 질문

그렇다면 프로덕트 개발에서 엔지니어 직무의 의미는?

Source: sushichan044, Giiku CAMP 아카데미아 발표(2026-07-08)

2.6 AI 코딩 확산 뒤 프로그래머의 역할과 경력

KEY INSIGHT 코드 작성과 설계 리뷰, 테스트까지 AI가 맡으면서 사람에게 남는 일이 불분명해짐 — 소닉가든은 그 답으로 '놀듯이 일하기'를 제시

AI가 맡는 작업

코드 작성
요구를 받아 코드를 생성

설계 리뷰
설계의 문제점을 검토

테스트
테스트 작성과 실행

프로그래머에게 남는 질문

- 내 역할은 무엇인가
- 내 경력은 어떻게 되는가

소닉가든의 답

'놀듯이 일하기'

개발을 AI에 넘기는 만큼 사람은 자기 일에 더 몰입하는 방향
이 방식을 15년간 사업으로 운영해 온 회사의 사례

Source: 구라누키 요시히토(소닉가든), Scrum Fest Sendai 2026 기조연설(2026.7.10)

2.6 지시만 처리하는 엔지니어의 경쟁 상대

KEY INSIGHT 지시 수행만으로는 쉬지 않고 일하는 AI를 이기기 어려움 — 과제 설정과 가치 탐색에 참여해야 역할이 남음

Q1 과제 설정과 가치 탐색은 PdM 만의 일인가

PdM과 엔지니어 모두 제품을 만드는 한 팀의 구성원

Q2 PdM은 생각하고 엔지니어는 만드는 분업으로 충분한가

생각과 구현을 나누면 엔지니어는 구현만 남음

Q3 지시받은 일만 하면 충분한가

남은 구현 업무는 AI가 가장 잘하는 영역

AI 에이전트의 작업 조건

- 불평 없이 24시간 작업
- 예산이 허락하는 만큼 지시 수행
- AI 에이전트에 맞춘 개발 프로세스 확산

지시 수행만으로는 이기기 어려움

걸림돌: 익숙한 방식을 버리는 일(언러닝)의 어려움

기존 기술에 머물면 새 기회를 얻지 못함 — 1차 산업혁명 때 숙련공이 겪은 위협과 같은 구조

Source: 원 발표 28~31쪽(오다카 도시유키, Kinoko Conference 2025)

질문별 답 줄은 원 발표의 문답을 요약

2.6 정형 코드 작성은 AI 몫으로 넘어감

KEY INSIGHT 사양에 직결되는 코드는 여전히 사람이 지시하거나 직접 씀 — 사람이 쓰는 코드는 줄었지만 한 줄의 중요도는 높아짐

AI가 맡는 코드

어떻게 구현해도 문제가 없는 코드

- **보일러플레이트**
- **테스트 코드**
- **다국어 처리**

이런 코드를 직접 쓰는 시간은 크게 줄어듦

사람이 맡는 코드

사양에 직결되는 부분 — 크게 편해지지 않음

- **자연어로 세세하게 지시**
  무엇을 어떻게 만들지 AI에 정확히 전달
- **직접 작성**
  지시보다 직접 쓰는 편이 빠른 부분

사람이 쓰는 코드 한 줄의 중요도는 높아짐

직접 코딩 스킬은 계속 필요하지만, 업무에서 차지하는 비중은 분명히 줄어듦

발표 요지 — 줄어든 비중만큼 사양 이해와 지시 능력의 중요도가 높아짐

Source: 원 발표(스리세이크, 2026.02.04) 일부 코딩 쪽

카드의 보조 설명은 보충

SECTION 2.7

개발자에게 남는 역량

판단력·코드 읽기·담당 범위 확장

KEY INSIGHT 사람 사이의 책임과 합의, 해결 방법을 찾는 판단은 사람에게 남음 — 엔지니어가 시간을 더 써야 할 자리가 여기로 좁혀짐

사람 대 사람이라서 남는 영역

의사결정

무엇을 할지 정하고 결과를 책임짐

커뮤니케이션

기획·디자인·사업 부서와 합의를 만듦

언어 모델이 해내지 못해 남는 영역: 지적 생산

제품 과제의 해결 방법 도출

어떤 문제를 어떤 방식으로 풀지 찾아냄

UI 설계

사용자 화면 구조

코드 설계

모듈·구조 결정

그 밖의 항목

발표 자료는 줄임표로 표시

보충 결과에 대한 책임과 이해관계자 합의는 도구에 위임하기 어려워 엔지니어가 시간을 더 써야 할 자리로 남음

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 7쪽

영역 구분과 항목 이름은 원 발표 기준. 항목별 한 줄 설명과 하단 띠는 보충

KEY INSIGHT 반복 작업은 AI가 점점 더 빠르게 처리함 — 결과를 확인하고 판단하는 능력이 엔지니어의 가치를 정함

맡길 일 — AI와 함께 빠르게

- **만들기·시험** 시제품·검증 환경 구축
- **초안 작성** 설계서·절차서·IaC
- **1차 리뷰** 누락 항목 지적
- **정형 운영** 로그 요약·1차 원인 분리
- **자료 수집** 조사·비교 자료 모으기
- **자동화** 반복 작업 자동화

키울 역량 — 사람이 책임

- **확인** 무엇이 정말 올바른지 판별
- **검증** 근거를 직접 확인
- **판단** 영향·위험·비용을 따져 결정
- **통합** 아키텍처·거버넌스 정합성 유지
- **장애 대응** 원인 규명·재발 방지
- **육성** 다음 세대 엔지니어 키우기

Source: 원 발표 p.29–30

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

KEY INSIGHT AI는 기술을 배우는 능력도 증폭함 — 그런데도 많은 엔지니어가 다른 역량을 택해 기술력 격차가 유지됨

발표 자료의 결론

AI 시대에도 높은 기술력에는 희소성이 있음

근거: AI는 능력을 증폭하는 도구
기반 기술력의 차이가 결과의 차이로 남음

단서

그 기술력에 가치가 있는지는
사용자가 있는지로 따로 따져야 함

가능성: AI로 기술력을 키울 수 있음

AI를 학습에 쓰면 기술력 향상도 가능

AI는 기술을 배우는 능력도 증폭함

예: 낯선 코드베이스 해설, 개념 질의, 구현 비교

현실: 다른 역량에 집중

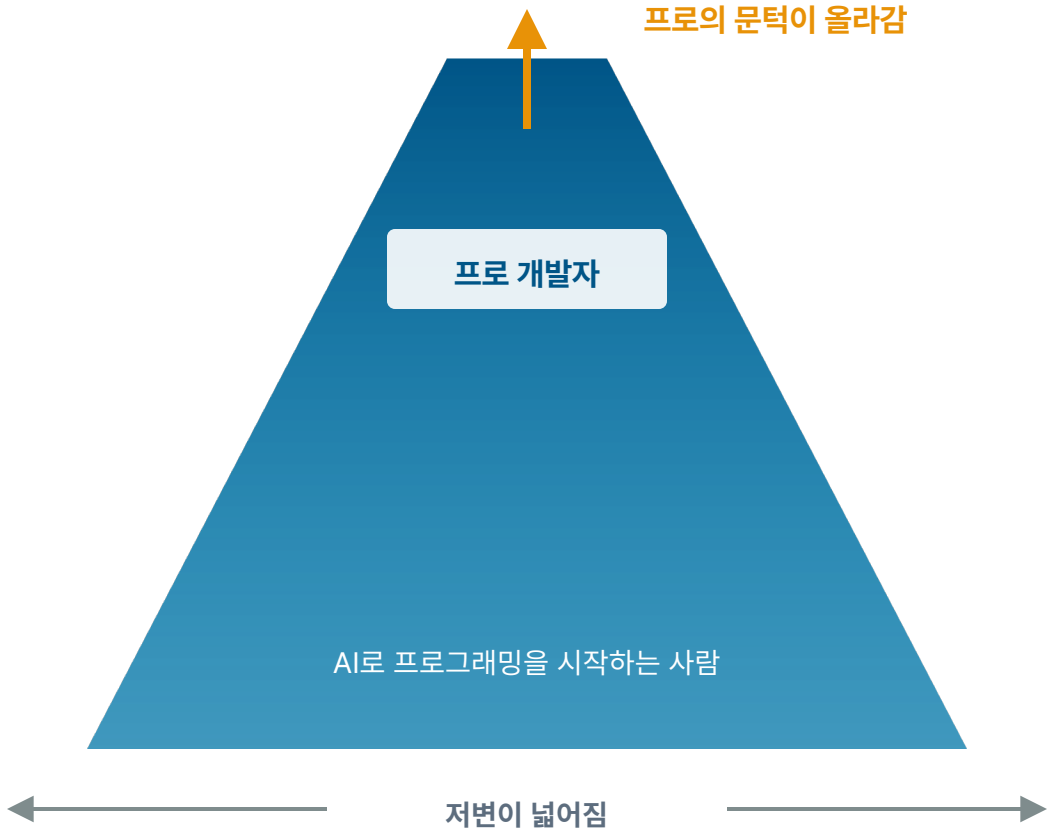
프로젝트 관리 등 기술 외 역량 개발에
힘을 쏟는 엔지니어가 많음

결과적으로 높은 기술력의 희소성이 유지됨

Source: 원 발표 12·13쪽(uhyo, 카오나비, 2026.06.18). 학습 활용 예시는 보충

AI 시대에 프로 개발자의 문턱은 높아짐

KEY INSIGHT 혼자 할 수 있는 일의 폭이 넓어져 무엇이 좋은지 가려내는 심미안이 기준이 됨 — 입문 저변은 넓어지지만 프로로 인정받기는 더 어려워짐



일은 사라지지 않음

AI로 혼자 할 수 있는 일의 폭이 넓어짐

적은 인원으로 더 큰 소프트웨어

소수 팀이 만들 수 있는 규모가 커짐

중심은 여전히 소스 코드

코드의 좋고 나쁨을 판단할 사람이 필요함

요구되는 역량

심미안

무엇이 좋은 소프트웨어인지 가려내는 눈

Source: 구라누키 요시히토(소닉가든), Scrum Fest Sendai 2026 기조연설(2026.7.10)

KEY INSIGHT AI가 쓴 코드는 읽지 않고 그대로 PR로 낼 수 없음 — 코드를 쓰는 비중은 줄어도 읽는 비중은 커짐

AI에 코드를 쓰게 했을 때

구현 장면

그대로 PR로 낼 수 없음

완성하려면 먼저 읽고 이해해야 함

AI에 코드를 조사하게 했을 때

조사 장면

그 결과가 정말 답인지 확인해야 함

근거가 되는 코드를 직접 따라가 봄

일반화하면 AI 출력 검증이 필요하고, 엔지니어는 그 일부로 코드를 읽을 수 있어야 함

보충: 개발자 설문 2025

AI 출력 정확도에 대한 태도

46% 정확도를 불신

33% 정확도를 신뢰

가장 큰 불만

66% 거의 맞지만 조금 틀린 답

겉보기에 맞는 코드일수록
읽어서 확인하는 단계가 필요함

Source: 원 발표(스리세이크, 2026.02.04) 코드 읽기 쪽; Stack Overflow Developer Survey 2025 AI 항목 (survey.stackoverflow.co/2025/ai)

오른쪽 설문 수치는 보충이며 원 발표에 없음

KEY INSIGHT

직접 설계한 것을 운영해 봐야 설계의 답을 맞춰 볼 수 있음 —
풀사이클 엔지니어는 여러 팀을 오가며 이 경험을 쌓기 쉬움

판단력: 엔지니어의 가치가 옮겨 감

쓰는 일 → **판단하는 일**

- 구현·디자인 작업 자체는 AI가 빨라지게 함
- AI의 답을 그대로 믿지 않고 좋고 나쁨을 가리는 능력
- 품질과 결과에 책임지는 판단력이 인재의 기준

담당 범위: 넓을수록 판단력이 오름

AI에 물어 가며 인접 영역으로 진입

- 인프라·모니터링·장애 조사처럼 전공 밖이던 일
- 몇 년 걸리던 인접 영역 학습 기간이 짧아짐
- 대응할 수 있는 범위가 넓은 사람이 더 중요해짐

키우는 방법은 경험을 계속 쌓는 것뿐

- 설계만, 운영만 해서는 판단력이 붙지 않음 — 직접 설계하고 만든 것을 운영해 봐야 설계를 검증할 수 있음
- 이 왕복을 여러 현장에서 반복해야 함
- 한 분야 전문가는 옮겨 갈 팀이 적지만, 풀사이클 엔지니어는 어느 팀에도 합류 가능(Dynamic Reteaming)

→ 다양한 프로젝트에 참여하며 경험을 더 빨리 쌓음

Source: 원 발표 자료 83~84쪽

원 발표는 스티븐 코비의 성공하는 사람들의 7가지 습관을 인용해 성장 단계는 건너뛸 수 없다고 설명함(요지)

SECTION 2.8

개발 현장의 생산성 실제

AI 작성 코드 비중과 신뢰 격차, 실제 단축 사례

KEY INSIGHT AI 결과를 검토할 판단력이 병목이 됨 — 한국 개발 조직에도 사고 훈련이 필요한 같은 이유

AI 도구 사용·사용 계획

84%

▲ 2024년 76%에서 상승

개발 과정에 AI 도구를 쓰거나
쓸 계획인 응답자 비율

AI 결과 정확성 불신

46%

▲ 2024년 31%에서 상승

AI 도구 결과의 정확성을
믿지 않는다는 응답 비율

가장 큰 불만

66%

"거의 맞지만 조금 틀린 답"

그럴듯한 오답을 걸러 낼
판단력이 필요함

발표 자료와의 연결

AI 사용이 늘수록 결과를 고르고 검토하는 일이 커짐 → 발표 자료가 말한 수직적 발달(판단의 틀)이
개발 현장의 실제 수요로 나타남

Source(보충): Stack Overflow 2025 Developer Survey 보도자료, stackoverflow.co/company/press/archive/stack-overflow-2025-developer-survey

코드의 약 95%를 AI가 작성함

KEY INSIGHT 코드 작성 시간이 크게 줄어듦 — 남는 시간을 어디에 쓰는지가 엔지니어의 일을 정함

AI가 작성한 코드 비율

발표자 업무 기준, 대략값

약 95%



직접 작성 약 5%

코드 작성은 AI에 맡기고
사람은 무엇을 만들지 판단

신입 연수

프로덕트 기능 개발

고객이 쓰는 기능을 기획부터 출시까지 담당

코드의 약 95%를 AI가 작성

현재 업무

플랫폼 통합 기술 과제

STORES 서비스를 하나의 플랫폼으로 통합하는 과제

여기서도 코드의 약 95%를 AI가 작성

Source: sushichan044, Giiku CAMP 아카데미아 발표(2026-07-08) — 95%는 발표자 개인 업무 기준의 대략값

라이브러리 원인 조사가 며칠에서 1시간으로 줄어듦

KEY INSIGHT AI가 코드를 읽어 원인 후보를 찾아 줌 — 그 후보가 진짜 원인인지 확인하는 일은 여전히 사람이 맡음

생성형 AI 이전

라이브러리 동작이 원인인 버그

① 구글 검색 → 원인을 찾지 못함

② 라이브러리 공식 문서 확인 → 찾지 못함

③ 라이브러리 이슈 검색 → 찾지 못함

④ 라이브러리 구현을 직접 읽어 해당 부분 발견

⑤ 대응 방법 검토·수정, 리뷰에서 설명

여기까지 걸린 시간

며칠

생성형 AI 이후

정기 회의 중 고객 문의

① 라이브러리 기능 문의를 받고 조사 약속

② 라이브러리를 클론해 AI에 조사 요청

③ AI가 원인 후보 위치를 찾아냄

④ 정말 그것이 원인인지 사람이 확인

⑤ 답변 문안도 AI로 작성해 전송

여기까지 걸린 시간

1시간

Source: 원 발표(스리세이크, 2026.02.04) 생성형 AI 이전·이후 업무 장면 쪽

발표자 개인 업무 사례 한 건이며 일반적인 단축 비율이 아님

KEY INSIGHT 처리한 양과 질이 그대로 개인 성과로 드러남 — 다만 몇 배까지 맡을지는 신중하게 정해야 함

출발점이 되는 질문

AI가 일한 몫을 어떻게 내 성과로 평가받을 것인가

2~3인분

업무를 더 많이 맡음

혼자서는 처리할 수 없던 양의
작업을 스스로 가져옴

AI와 함께 처리

작업은 AI에 맡기고
사람은 지시와 확인을 맡음

처리량과 품질이 성과로 보임

AI의 몫이 따로 떨어지지 않고
맡은 사람의 결과로 남음

주의: 몇 배까지 맡을지는 신중하게

배수는 개발 방식과 AI 중단 위험에 따라 달라짐. 처음부터 높게 약속하면 되돌리기 어려움

Source: 원 발표(TRAILBLAZER Kazuhisa Naoi, 2026.08.21) 성과 평가 쪽

2~3인분 업무 수행은 원 발표가 제시한 방법. 카드 설명 줄과 주의 문단 둘째 줄은 보충

KEY INSIGHT 애자일은 스프린트마다 배수를 조정할 수 있지만 워터폴은 장기 일정이 변화에 약함 — 워터폴은 2배 정도가 현실적임

내재화 × 애자일

맡을 수 있는 배수

2~3인분

- 2인분·3인분으로 넉넉하게 맡을 수 있음
- 스프린트 단위로 조정하기 쉬움
- 실적을 보며 배수를 조금씩 올릴 수 있음

짧은 주기로 계획을 고칠 수 있어 초과 약속의 위험이 작음

워터폴

현실적인 배수

2배 정도

- 몇 배를 맡는 것 자체는 가능함
- 장기 일정이 변화에 약함
- 안전 여유를 두면 2배 정도가 현실적

계획이 길게 묶여 있어 배수를 잘못 잡으면 뒤늦게 드러남

Source: 원 발표(TRAILBLAZER Kazuhisa Naoi, 2026.08.21) 개발 방식별 배수 쪽

배수는 원 발표자의 경험에 근거한 판단이며 측정값이 아님. 각 열 맨 아래 해설 줄은 보충

KEY INSIGHT 반복 구현은 AI로 넘어가고 명세 작성과 결과 검증이 핵심 업무가 됨 — 생산성 보고도 체감 속도에서 명세 품질·검증 결과로 바꿔야 함

사라지는 일	남는 일	새로 생기는 일
반복 구현과 정형 코드 작성	아키텍처 판단과 설계 검토	에이전트가 따를 명세(요구사항, 완료 조건) 작성
단순 테스트 코드 작성	장애 원인 분석	AI가 만든 변경의 검증과 회귀 확인
문서·주석 초안 작성	운영 안정성에 대한 책임	개발 에이전트가 참조할 코드 규칙·자료 관리

에이전트가 만든 변경마다 명세를 쓴 사람과 승인한 사람을 기록하고, 보안 검토가 필요한 변경은 사람 승인 없이 배포되지 않게 막음

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.4.1절(2026.10) · METR(2025.07)

SECTION 2.9

디자인 업무의 변화

제작 시간이 줄고 검증 시간이 늘어난 디자인

KEY INSIGHT

속도 향상을 느끼는 비율보다 품질 향상을 느끼는 비율이 낮음

빨라진 만큼 검증 업무가 늘어남

발표 자료

가장 큰 영향은 제작 속도

IT 제작 현장의 화면 · 시제품을
매우 빠르게 만들 수 있게 됨

화면과 시제품이 빨리 나오면서
만드는 순서와 직군 역할이 함께 바뀜

보충 · Figma 2025 AI Report (디자이너 · 개발자 약 2,500명)

작업 흐름이 크게
빨라졌다는 응답

78%

속도 체감

작업 품질이
좋아졌다는 응답

58%

품질 체감

두 응답의 차이 20%p

빠르게 나온 결과물의 품질은 사람이 따로 확인해야 함

Source: 원 발표(2026-06-18); 보충: Figma 2025 AI Report, 디자이너·개발자 약 2,500명 조사 (figma.com/reports/ai-2025, figma.com/resource-library/design-statistics)

AI 활용 온도차: 엔지니어와 디자이너

KEY INSIGHT 엔지니어는 이미 일하는 방식이 바뀌었고 디자이너는 개인차가 큼 — 같은 조직 안에서도 도입 속도를 직군별로 따로 잡아야 함

엔지니어

변화가 이미 진행 중

- AI 코딩 지원으로 일하는 방식이 이미 바뀜
- 새 기술에 대한 심리적 장벽이 낮음
- 대응하지 않으면 뒤처진다는 위기감을 팀 안에서 공유함

디자이너

개인차가 큰 상태

- AI 활용이 아직 발전 단계이고 받아들이는 온도가 사람마다 크게 다름
- 도구 교체 경험이 비교적 적어 변화에 대한 인식이 제각각임
- 그래픽 출신 등 배경이 다양해 생각의 차이가 생김

도입 계획의 함의(보충)

같은 도구라도 디자이너 쪽은 개인별 숙련도 차이를 전제로 교육과 적용 범위를 나눠 잡는 편이 안전함

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 7쪽

두 직군 비교는 원 발표자의 관찰이며 조사 수치가 아님. 하단 함의는 보충

업무 흐름 변화의 영향은 UI 디자이너가 더 큼

KEY INSIGHT

UI 제작은 자동 생성과 구현 연계가 깊어 직접 영향을 받음

UX 설계의 핵심 사고는 아직 AI로 대체하기 어려움

UX 디자이너

영향 작음

설계 영역(사고)

사용자 경험의 핵심을 정의하는 사고 과정은
AI로 대체하기 아직 어려워 영향이 제한적

구현과 떨어진 상류 과정

구현과 거리가 있어 다른 직군에 주는 영향이
적고, 업무 흐름 변화의 직접 영향도 적음

AI 활용 방식

일부 작업을 돕는 지원 도구로 사용

UI 디자이너

영향 큼

제작 영역(생성 · 산출물)

자동화가 가장 빠르게 진행되는 영역이라
AI 자동 생성의 효과를 바로 받음

엔지니어와의 연계(구현)

구현 과정과 깊게 연결되어 있어
업무 흐름 전체가 AI로 크게 바뀜

영향 크기 비교(발표 자료)

UX 디자이너 < UI 디자이너

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

KEY INSIGHT

처음부터 목업을 만드는 일은 줄어듦

도구 비중은 AI 도구 70%, 디자인 도구 30% 정도로 예상됨(발표 자료 추정)

기존 UI 디자인 업무 흐름

1 디자이너가 디자인 도구로 시제품 제작

2 디자이너가 로파이(Lo-Fi) 시제품을 만들고 엔지니어가 구현

목업을 처음부터 사람이 직접 그림

AI 시대 UI 디자인 업무 흐름

1 AI 도구로 시제품 제작

2 디자이너 · 엔지니어가 함께 검증 · 수정 품질을 높임

처음부터 목업을 만드는 일은 줄어듦

UI 디자이너의 도구 사용 비중

발표 자료 추정, 실측값 아님



AI 도구: 시제품 제작

디자인 도구: 다듬기 + 핸드오프 자료 작성

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

KEY INSIGHT

디자이너 업무가 형태 제작에서 제품 가치 판단 쪽으로 옮겨 감

수치는 발표 자료의 예시 이미지 기준임

형태로 만드는 시간이 줄고 제품 가치를 다듬는 시간이 늘어남



사고 · 검증

30→60%

비중이 두 배

줄어드는 시간

화면을 형태로 만드는 시간

목업 · 시제품 제작은 AI 도구가
상당 부분을 맡음

늘어나는 시간

제품 가치를 다듬는 사고 · 검증 시간

디자이너의 일이 사고하는 부분에
집중되는 방향으로 바뀜

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

원 발표 주석: 그래프 내용은 이미지 예시이며 실측값이 아님

KEY INSIGHT 정답이 있는 반복 작업은 AI가 더 잘함 — 엔지니어는 이미 체감하지만 디자이너의 위기감은 아직 낮음

픽셀 단위 미세 조정

간격 · 정렬 · 크기를 손으로 맞추는 작업

단순한 컴포넌트 코딩

정해진 모양의 버튼 · 카드 · 목록을 코드로 옮기는 작업

반복되는 양산 작업

비슷한 화면 · 배너 변형을 여러 벌 만드는 작업

세 작업의 공통점

구체적이고 정답이 있는 영역

결과를 기준과 맞춰 볼 수 있어
AI가 빠르고 정확하게 처리함

직군별 체감 차이

엔지니어	대부분 이미 변화를 체감
디자이너	위기감이 아직 충분하지 않음

발표 자료자의 관찰이며 조사 수치는 아님

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 12쪽

각 항목 아래 설명 줄은 보충 예시

SECTION 2.10

디자이너 역할과 역량

경험 설계자로 넓어지는 디자이너 역할

UI Creator와 Experience Architect의 차이

KEY INSIGHT 일의 중심이 구체 작업에서 추상 전략으로 옮겨 감 — 최종 사용자 경험의 책임까지 디자이너가 맡게 됨

	UI Creator	역할 진화	Experience Architect
	화면을 만드는 사람	→	사용자 경험을 설계하는 사람
작업 수준	구체적인 작업	→	**더 추상적인 전략**
도구와의 관계	도구를 다루는 운영자	→	**시스템 전체를 이끄는 지휘자**
책임 범위	결과물을 넘겨주는 데서 끝	→	**최종 사용자 경험에 대한 오너십**

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 11쪽

왼쪽 분류 이름(작업 수준 · 도구와의 관계 · 책임 범위)은 이해를 돕기 위한 보충

KEY INSIGHT 세 역량 모두 AI에게 의도를 정확히 넘기는 데 쓰임 — AI 동작 원리를 모르면 세 역량이 결과로 이어지지 않음



추상적 사고력

사물의 본질을 파악하는 힘

화면 여러 장에서 공통 규칙을 뽑아
디자인 시스템으로 정리



미래 사고

변화를 예측하고 계획하는 힘

개인화 · 동적 UI처럼 다가올
환경을 미리 설계에 반영



언어화 능력

AI와 대화하며 의도를 전달하는 힘

디자인 의도와 규칙을 프롬프트 ·
규칙 문서로 명확히 적음



세 역량의 기반

AI에 대한 깊은 지식

AI가 무엇을 잘하고 어디서 틀리는지 알아야 의도를 정확히 넘길 수 있음

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 13쪽

카드 아래 회색 적용 예와 기반 띠의 오른쪽 해설은 보충

KEY INSIGHT 실현 가능한 설계로 목적을 이루는 책임은 도구가 AI로 바뀌어도 그대로임 — AI는 그 책임을 수행하는 수단

발표 자료가 제시한 디자인의 정의(요지)

어떤 목적을 이루기 위해
실현 가능한 설계와 계획을 하는 일

바뀌는 것: 도구와 작업 방식

그리는 도구가 AI로 바뀌고, 픽셀 조정과
단순 코딩은 AI가 맡음

그대로인 것: 디자이너의 책임

목적에 맞고 실제로 만들 수 있는 설계를
책임지는 한 디자인의 본질은 같음

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 14쪽

가운데 정의는 원 발표 14쪽 내용의 요지를 한국어로 다시 쓴 것

KEY INSIGHT

손으로 만드는 일을 AI가 대신해도 이런 질문은 그대로 사람 몫으로 남음

품질 기준을 정하는 주체는 여전히 사람임

디자이너는 손을 움직이는 동안 아래 질문을 동시에 확인하며 작업함

접근성

접근성 대응이 되었는가

정보 구조

정보 우선순위가 바르게 정리되었는가

시선 흐름

시선 흐름이 자연스럽게 설계되었는가

일관성

전체에서 볼 때 이 동작이 일관된가

콘셉트

서비스 콘셉트에 맞는 외형인가

배치

이 배치로 사용자가 곤란하지 않은가

디자인 원칙

기본 디자인 원칙을 따르는가

예외 상황

예외 상황에 대응하는가

기능 범위

용도를 보면 이 기능도 필요하지 않은가

손 작업을 AI가 대신하면

손으로 만드는 일은 AI가 대신하지만 이 질문들은 모두 사람 몫으로 남음

AI가 만든 화면에도 같은 질문 적용 그래야 품질 기준이 유지됨

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

KEY INSIGHT

실제 제작과 기본 원칙 준수는 AI가 맡고 브랜드·이용 맥락 최적화와 의사결정은 디자이너가 맡음

차별화는 제품 고유 맥락에서 나옴

AI가 일반적인 정답을 대량으로 만들면서, 디자이너는 제품 고유 맥락의 가치를 만드는 일에 집중함

구분	업무	기존 역할	AI 시대 역할
제작	**실제 작업**	디자이너	AI가 대신함
사고	**기본 디자인 원칙 준수**	디자이너	AI가 대신함
사고	**브랜드에 맞춘 최적화**	디자이너	**디자이너**
사고	**이용 맥락에 맞춘 최적화**	디자이너	**디자이너**
확인	**확인 · 검증**	디자이너	**확인 · 검증 + 의사결정**

기존 일하는 방식

손을 움직이며 탐색하고, 생각하고, 구체화

AI 시대 일하는 방식

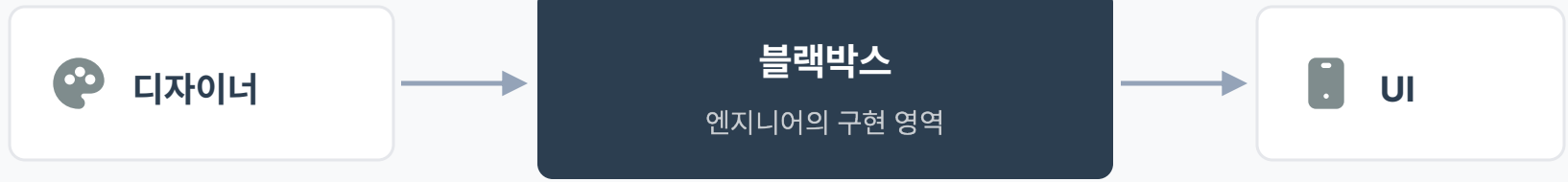
AI를 움직이며 탐색하고, 정의하고, 최적화

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

2.10 디자이너가 구현 결과를 직접 제어하게 됨

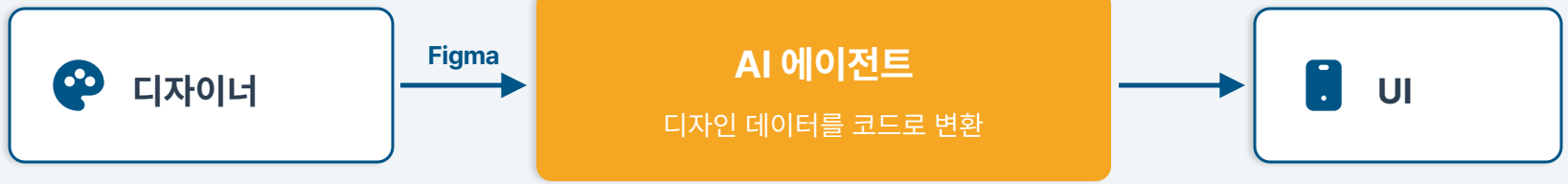
KEY INSIGHT AI 에이전트가 엔지니어 손에 있던 변환 구간을 맡음 — 최종 화면 품질이 디자인 데이터 품질에 직접 묶임

Before



디자이너가 손댈 수 없음

After



디자이너가 직접 제어

발표 자료의 평가: 부정적인 변화로 볼 일은 아님

손댈 수 없던 구현 영역을 Figma로 제어하게 되어, 사용자에게 가는 최종 결과물에 디자이너가 직접 다가갈 수 있음

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 9쪽

SECTION 2.11

AI가 읽는 디자인 시스템

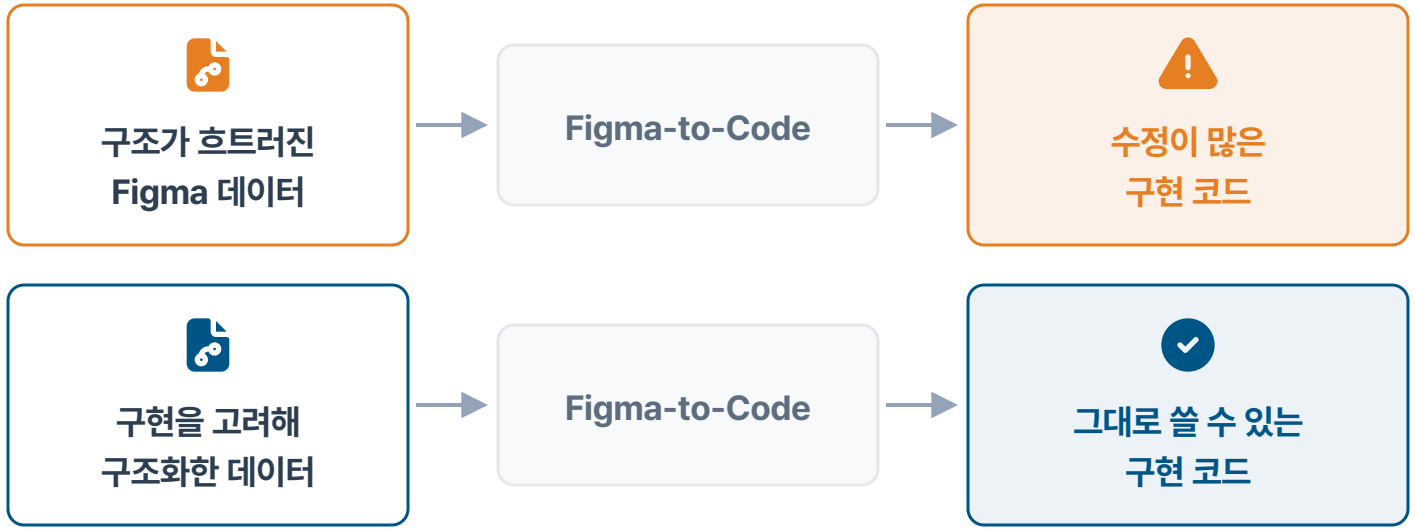
Figma-to-Code 와 AI용 디자인 시스템

2.11 Figma-to-Code 품질은 입력 데이터가 결정함

KEY INSIGHT 완벽한 데이터를 디자이너 혼자 계속 보장하기는 어려움 — 그래서 점검 도구가 코드 생성보다 먼저 필요함

Garbage In, Garbage Out

질 낮은 데이터를 넣으면 질 낮은 코드만 나옴



흐트러진 데이터의 예(보충)

- Auto Layout 없이 좌표로만 놓은 요소
- 컴포넌트가 아닌 복사본, 이름 없는 레이어
- 디자인 토큰 대신 임의 색상·수치
- 상태(눌림·비활성) 화면 누락

디자이너 혼자 지는 부담

늘 구현을 의식한 완벽한 데이터를
디자이너만으로 계속 보장하기는
매우 큰 부담

발표 자료의 결론

코드 생성 정확도를 올리려면
입력 데이터를 자동으로 점검하는
Figma Linter가 먼저 있어야 함

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 4쪽

흐트러진 데이터의 예는 원 발표에 없는 보충 예시

2.11 단기 변화: 비주얼 제작보다 구조 설계 중시

KEY INSIGHT AI가 해석할 수 있는 구조적 디자인 데이터가 필요해짐 — 모호하게 넘기던 설계가 곧바로 문제로 드러남

한 장짜리 비주얼



겉모습은 같아도 규칙 없이
좌표로만 놓인 요소

AI가 의도를 해석하기 어려움

구조화된 디자인 데이터

- 화면 프레임
  - Auto Layout 영역
    - 컴포넌트 · 변형
    - 상태(기본·눌림·비활성)
  - 디자인 토큰(색 · 간격 · 글자)

계층 · 규칙 · 상태가 데이터에
담겨 그대로 코드로 바뀜

기계적 변환에 맞는 형태

구조적 데이터의 중요성 확대

AI가 해석하는 기계적 변환에 맞게 만든
디자인 데이터가 매우 중요해짐

UI 디자이너의 책임은 화면 설계

화면이 동작하는 구조를 정하는 일이 본업.
한 장짜리 비주얼은 그 결과물의 일부

모호한 설계는 더 이상 허용되지 않음

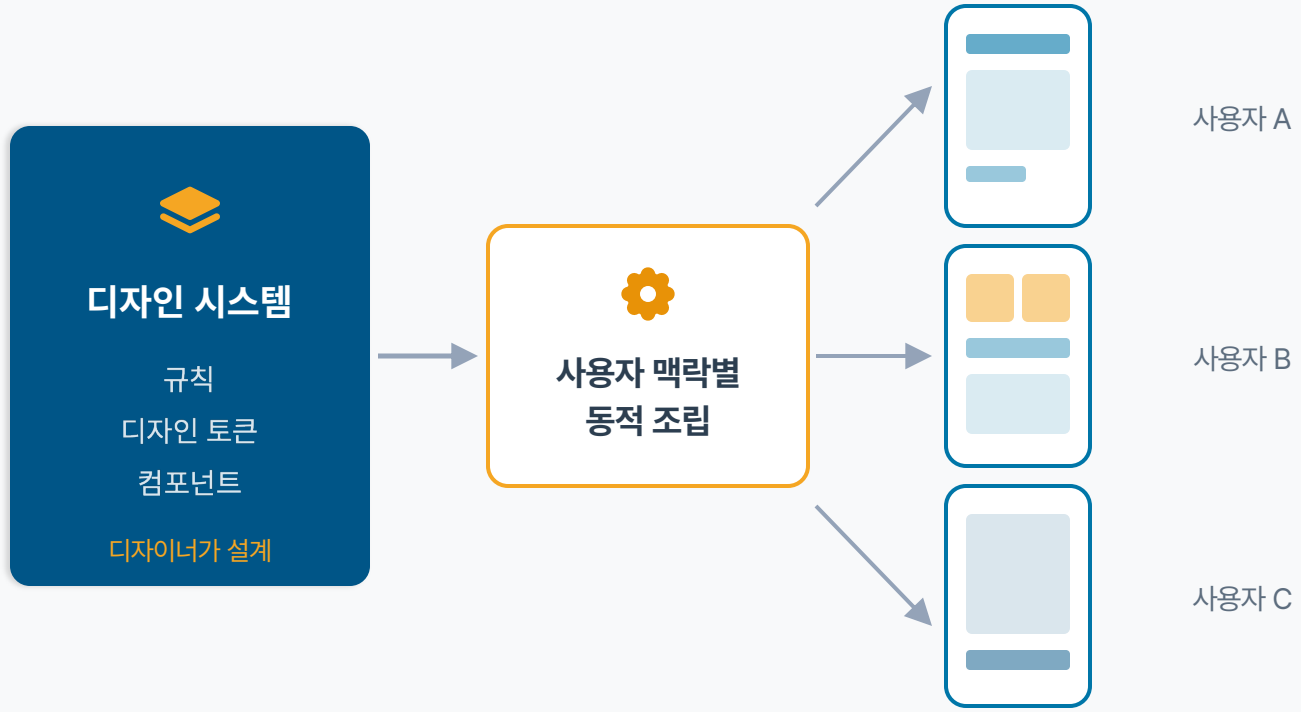
사람이 눈치로 메우던 빈칸이 생성 코드에
그대로 드러나 문제로 표면화됨

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 8쪽

왼쪽 두 그림(비주얼과 계층 구조 비교)은 원 발표 내용을 설명하기 위해 덧붙인 보충 도식

2.11 중기 변화: 개별 화면보다 규칙과 시스템

KEY INSIGHT UI가 사용자 기기에서 동적으로 개인화되면 화면을 한 장씩 그릴 수 없음 — 디자인 시스템 구축이 디자이너의 주 업무가 됨



같은 규칙에서 사용자마다 다른 화면이 만들어짐

발표 자료의 전망

UI 개인화가 사용자 기기 안에서 이뤄지는 환경이 옴

디자이너에게는 디자인 시스템 자체를 구축하는 역할이 강하게 요구됨

보충: 생성형 UI(Generative UI)

AI가 사용자 목적과 맥락에 맞춰 화면을 그때그때 조립하는 방식

디자이너는 목표와 제약 조건을 정하고 AI가 그 안에서 화면을 구성함

업계 논의 단계, 일반화된 표준은 아님

Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 10쪽, foolproof.co.uk 'What is Gen UI and does it matter'

보충: 생성형 UI 설명과 왼쪽 흐름도는 원 발표에 없는 외부 자료·해설

2.11 AI 시대 UI 디자이너의 핵심 업무

KEY INSIGHT

AI 성능을 끌어낼 환경을 만들고 결과를 평가·통합하는 일이 중심이 됨

디자인 시스템이 AI 입력 자료가 됨

직접 만드는 비중보다 AI에 지시하는 비중이 커지면서 중요해지는 업무

디자인 시스템

사람용에서 AI용으로 갱신

사람이 보던 디자인 시스템을
AI가 읽기 쉬운 형식으로 갱신

새로 포함할 요소

skills · DESIGN.md 같은
AI용 정의도 디자인 시스템의 일부

컨텍스트 설계

자사 서비스에 맞춤

일반적인 AI 출력을 제품 고유
맥락과 지시로 서비스에 맞춤

얻는 효과

유사 제품과의 차별화,
제품 경쟁력 강화

검증과 판단

여러 기준으로 평가

AI 출력을 평가하고 수정을
지시하는 품질 관리

평가 기준

제품 고유 톤앤매너 · 접근성 기준
준수 · 사용자 과제 해결

보충 DESIGN.md: Google Stitch에서 나온 형식으로, 색 · 서체 · 간격 같은 디자인 토큰과 규칙을 AI 에이전트가 읽는 Markdown 파일 하나로 정리함(2026년 오픈소스 공개, 알파 단계)

Source: 원 발표(2026-06-18); 보충: The Decoder, Google open-source DESIGN.md (2026); MindStudio, What is DESIGN.md in Google Stitch

2.11 AI 지원 병렬 업무 흐름

KEY INSIGHT

기획 단계에서 시제품을 만들 수 있어 UX·UI·개발이 동시에 진행됨

직군 간 인수인계 대기가 줄어듦

기존 흐름

기획 → 디자인 → 구현을 바통 넘기듯 차례로 진행

UX 디자인
타깃 파악 · 과제 도출

→

UI 디자인
시제품 제작 ~ Hi-Fi 디자인

→

개발 · 구현
디자인을 받은 뒤 시작

AI 시대 흐름

기획 단계부터 시제품을 만들며 생각하는 병렬 진행

UX 디자인 타깃 파악 · 과제 도출

UI 디자인 시제품 제작 ~ Hi-Fi 디자인

개발 · 구현 초기 시제품부터 함께 검토

AI 지원: 전 구간에서 시제품 제작과 수정 속도를 높임

세 작업 구간이 겹쳐 진행되어, 앞 직군의 결과를 기다리는 시간이 줄어듦

Source: 와타나베 케이스케(Works Human Intelligence), WHI Developers 라이트닝 토크(2026-06-18)

2.11 AI-DLC 단계별 디자이너 검토 지점

KEY INSIGHT

Inception 에서는 UX, Construction 에서는 UI 관점의 리뷰가 필요함

직접 만드는 시간보다 리뷰·수정 시간이 길어짐

Inception
시작 단계

AI와 하는 일

만들 대상을 입력하고
서비스 정의 · 정책을 정함

디자이너: UX 관점 리뷰

- 타깃 설정이 어긋나지 않았는지
- 사용자 스토리 리뷰
- 리뷰하며 개발팀 인식 맞추기

Construction
구축 단계

AI와 하는 일

컴포넌트 작성과 코드 생성
단계마다 리뷰 · 수정에 무게

디자이너: UI 관점 리뷰

- UI 품질 리뷰
- 직접 만드는 시간보다
  리뷰 · 수정 시간이 더 김

Operation
운영 단계

AI와 하는 일

운영 배포와
배포 뒤 관리

디자이너 개입 적음

발표 자료에서 이 단계의
디자이너 검토 항목은
따로 제시되지 않음

Source: 원 발표(2026-06-18); AWS DevOps Blog, AI-Driven Development Life Cycle

SECTION 2.12

테스트·QA 직무의 변화

테스트 실행은 AI, 테스트 설계·판정은 사람

2.12 테스트 활동 6단계의 AI 적용 수준

KEY INSIGHT AI는 대부분의 활동에서 초안을 빠르게 내지만 품질 목표·도메인 관점·누락 확인은 여전히 사람이 맡음

테스트 활동	AI가 잘하는 것	사람이 보완할 것
테스트 계획	리스크 분석을 잘하고 테스트 레벨·타입을 사람보다 폭넓게 제시함	품질 목표는 사람이 정함 프로젝트 사정을 자세히 줘야 함
테스트 관점 도출	일반적인 테스트 관점을 폭넓게 냄	과거 이슈·CS 문의·암묵적 사양은 사람이 컨텍스트로 줌 모델에 따라 관점이 통째로 빠지기도 함
테스트 설계	경계값·대표값·결정 테이블 패턴을 냄	조합(페어와이즈)·상태 전이 패턴은 확인 필요 패턴이 통째로 빠진 사례가 있음
테스트 구현	테스트 코드 생성은 AI의 강점 설계가 있으면 사람용 케이스도 냄	사람용 절차가 실제로 실행 가능한지 확인 필요
테스트 실행	E2E 자동 테스트 구현·실행이 흔함 탐색적 테스트 에이전트도 MIXI에서 가동 중	사람용 케이스 실행 에이전트는 실험 단계
테스트 리포트	흩어진 정보 정리는 AI의 강점	그냥 만들면 모든 정보를 다 담으려 함 스킬·템플릿으로 형식을 고정

Source: 원 발표(MIXI 마쓰야 미네오, 2026.07.23) 테스트 활동별 AI 적용 소감 쪽

원 발표자 개인 견해를 요지로 정리함. 측정값이 아닌 현장 소감

2.12 수동 테스트 케이스 실행 에이전트 실험

KEY INSIGHT 102건 중 87건(85.2%)이 의도대로 실행됨 — 실행 정확도 지표라 버그 검출 성능과는 따로 봐야 함

의도대로 실행된 테스트

85.2%

실제 테스트 102건 중 87건

의도대로 실행 못 함 15건(14.8%)

사용 모델 GPT-5.5 (발표 자료 기준, 미확인)

실험 조건

사람용 수동 테스트 케이스를 그대로 실행하는 AI 에이전트를 MIXI가 개발

실제 테스트 102건으로 도구가 의도대로 동작하는지 확인

발표 자료자는 같은 행사 스폰서 세션에서 이 도구를 따로 소개함

수치를 읽는 법

측정 대상은 테스트를 의도대로 수행했는지이며 버그 검출 정확도가 아님

실행에 실패한 15건은 사람이 다시 실행하거나 케이스를 고쳐야 함

보충: 실패 15건의 원인 분류는 발표 자료에 없음

함께 공개된 자동 실행 스킬

테스트 계획·분석·설계·구현·실행·리포트를 차례로 부르는 스킬 묶음, Unit·Integration·E2E·사람 실행을 나눠 진행

공개 저장소 github.com/jam0824/ai_test_process_skills (구조는 22쪽)

Source: 원 발표(MIXI, 2026.07.23) 수동 테스트 실행 에이전트·자동 실행 스킬 쪽; github.com/jam0824/ai_test_process_skills

수치는 원 발표 기준. 저장소 공개 여부는 2026.10.29 확인

2.12 사람은 작업 설계와 결과 확인을 맡음

KEY INSIGHT 실행을 AI가 맡아도 요구가 틀리면 결과도 틀림 — 요구 내용과 확인 단계를 함께 설계해야 함

사람이 맡는 단계 / AI가 맡는 단계

1 요구 발생 → 2 작업 방침 결정 → 3 AI 분담 결정 → 4 작업 설계 → **5 작업 실행** → 6 산출물 타당성 확인 → 7 재사용 정리 등 완료 작업

발표 자료의 예: 테스트 계획서 작성

1 요구
테스트 계획서를 빨리 만들고 싶음

2 방침
리스크 기반, E2E 자동 테스트와 사람의 시스템 테스트를 계획

3 분담
일정은 미확정이라 사람이 넣음
리스크 도출·접근법은 AI가 맡음

4 설계
리스크 도출은 에이전트 A, 끝나면 에이전트 B로 리스크별 접근법 도출

요구를 정확히 냈는지도 확인 대상임. 요구가 모호하면 AI는 그럴듯하지만 다른 결과를 냄

실행이 사람에게 남을 수 있는 예외

정확도 미달
그 시점의 AI가 요구 정확도에 못 미치는 부분

미션 크리티컬
장애 영향이 큰 영역

사용감 확인
게임·앱을 직접 만져 봐야 아는 조작감

토큰 비용
AI로 돌리는 비용이 사람보다 큰 작업

Source: 원 발표(MIXI, 2026.07.23) 작업 방식의 재정의·사람의 움직임 쪽

7단계와 예외 4가지는 원 발표 기준. 예외 카드의 풀이 문구는 보충

2.12 AI 산출물에는 사람의 검토가 필요함

KEY INSIGHT AI는 사람이 하지 않는 종류의 실수를 함 — 사람 리뷰 또는 별도 리뷰 장치를 공정에 넣어야 함

사례 1 · 테스트 설계

한 번에 생성한 테스트 항목이 5건뿐

사양서·데이터·프롬프트를 준비함
관점 도출·설계·테스트 항목을 한 번에 생성

몇 분 만에 끝났지만 테스트 항목은 5건

이유를 묻자 AI는 이 규모에 비해 항목이 적다고 인정함

대량 생성을 한 번에 맡기면 일부가 빠질 수 있음

사용 모델 GPT-5.5 (발표 자료 기준, 미확인)

사례 2 · 긴 문서 생성

긴 글에서 앞뒤가 맞지 않는 서술

발표 자료자가 취미로 AI와 10장 분량의 소설을 생성

중간에 앞 장의 내용과 연결되지 않는
표현이 섞여 나옴

작은 실수보다 사람이라면 하지 않을 종류의 실수가 더 다루기 어려움

리뷰 공수가 늘어도 리뷰를 생략할 근거는 되지 않음

확인하지 않으면 요구대로인지 알 수 없음

사람 리뷰 또는 리뷰 장치를 공정에 두고, 요구를 정확히 냈는지도 함께 점검

Source: 원 발표(MIXI, 2026.07.23) 사람 확인의 중요성·실제 사례 두 쪽

두 사례는 원 발표자의 경험담을 요지로 정리함. 소설 본문 인용은 생략

2.12 테스트 엔지니어와 QA 엔지니어의 역할 재정의

KEY INSIGHT 두 역할 모두 사용자 만족을 실현하는 체계를 만드는 일로 모임 — 실현 수단이 수동 테스트에서 AI로 바뀌어도 목표는 그대로임

테스트 엔지니어

제품 관점

테스트 활동에 집중

개발 프로세스 전체의 테스트 활동을 정비하고
체계화함

하는 일의 예

전 테스트 레벨을 대상으로 계획·분석·설계

Unit·Integration·E2E 구현과 사람용 항목 생성

실행 결과를 리포트로 모으는 파이프라인 구축

QA 엔지니어

프로젝트 관점

품질 보증 활동에 집중

규칙·프로세스 같은 환경을 정비하고 교육해
프로젝트 전체가 품질 보증 활동을 하게 함

하는 일의 예

사람·AI를 포함한 프로젝트 규칙과 프로세스 설정

AI 사용법과 가드레일 정비

구성원 대상 품질 보증 교육과 육성

공통 목표: 사용자 만족을 실현하는 체계 구축

목표는 예전과 같고, 수동 테스트로 실현하던 것을 이제 AI를 포함한 체계로 실현함

Source: 원 발표(MIXI, 2026.07.23) 역할의 재정의·우리 일의 재정의 쪽

두 역할의 정의는 원 발표의 요지

2.12 테스트 엔지니어의 필수 역량

KEY INSIGHT 지식이 없으면 AI에 낼 요구도, 결과를 판정할 기준도 없음 — AI가 작업할수록 테스트 기초 지식의 비중이 커짐

테스트 범위의 재정의

테스트는 시스템 테스트 레벨에 한정되지 않고 개발 프로세스 전체의 테스트 활동을 가리킴

테스트 기초 지식

요구를 내는 출발점

API 테스트를 모르면
API 응답으로 확인하자는
요구 자체를 낼 수 없음

산출물 평가

판정 기준을 머리에 둠

기준이 없으면 AI 출력이
다 맞다고 믿고 리뷰가
필요 없다고 여기게 됨

의도 전달·독해

지시를 쓰고 답을 읽는 힘

주어·목적어가 빠진
지시는 의도가 전달되지
않고, 답을 못 읽으면
오류를 놓침

AI 활용 능력

당연히 요구되는 기본기

오피스 도구를 쓰는
수준처럼 요구되며 현재
도구로 할 수 있는 일을
알아 둠

프로세스 재설계

도구에 맞게 공정을 바꿈

비효율적인 기존 방식을
검토 없이 그대로
자동화하지 않음

Don’t pave the cow path
(Hammer, HBR 1990)

Source: 원 발표(MIXI, 2026.07.23) 테스트 기초의 재정의 쪽; Hammer, ‘Reengineering Work’, HBR 1990

다섯 역량은 원 발표 기준. Hammer 원전 표기는 보충

SECTION 03

소프트웨어 개발 방식의 변화

에이전트 지시·작은 팀·소프트웨어 팩토리·확인 지점·하네스·부채·가치 측정

SECTION 3.1

에이전트에 작업을 맡기는 개발

작업 단위로 맡기는 코딩 에이전트

KEY INSIGHT AI로 만드는 속도는 빨라졌지만 팀은 그 속도를 따라가느라 바쁨 — 리뷰 부담과 특정인 의존이 늘어남

2021년의 소프트웨어 개발

개발자가 프로그램을 작성

코드를 직접 쓰고 고치는 일이 중심

5년

2026년의 소프트웨어 개발

개발자가 프롬프트를 작성

AI에게 줄 지시문을 쓰고 결과를 확인

현장에서 생긴 일

리뷰 부담 증가

AI가 만든 변경을
사람이 다 읽어야 함

대화 감소

분업이 늘어 사람 사이
의사소통이 줄어듦

특정인 의존 가속

맡은 사람만 아는
코드와 맥락이 늘어남

이해 지연

변경 속도를 팀의
이해가 따라가지 못함

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07). 현장 사례는 원 발표자의 관찰

KEY INSIGHT 가치라는 목적은 같고 동작을 지정하는 방법만 바뀜 — 프로그래밍 숙련도보다 요구를 정확히 전하는 능력이 중요해짐

기존 방식

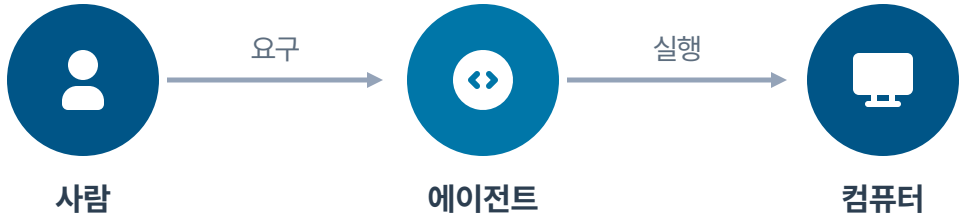
사람이 동작을 직접 규정



코드로 컴퓨터 동작을 한 줄씩 지정

새 방식

에이전트가 동작을 대신 구성



사람의 요구를 받아 에이전트가 컴퓨터를 움직임

변하지 않는 것: 가치(본질)

누구에게 어떤 가치를 전할지는 두 방식 모두 같음

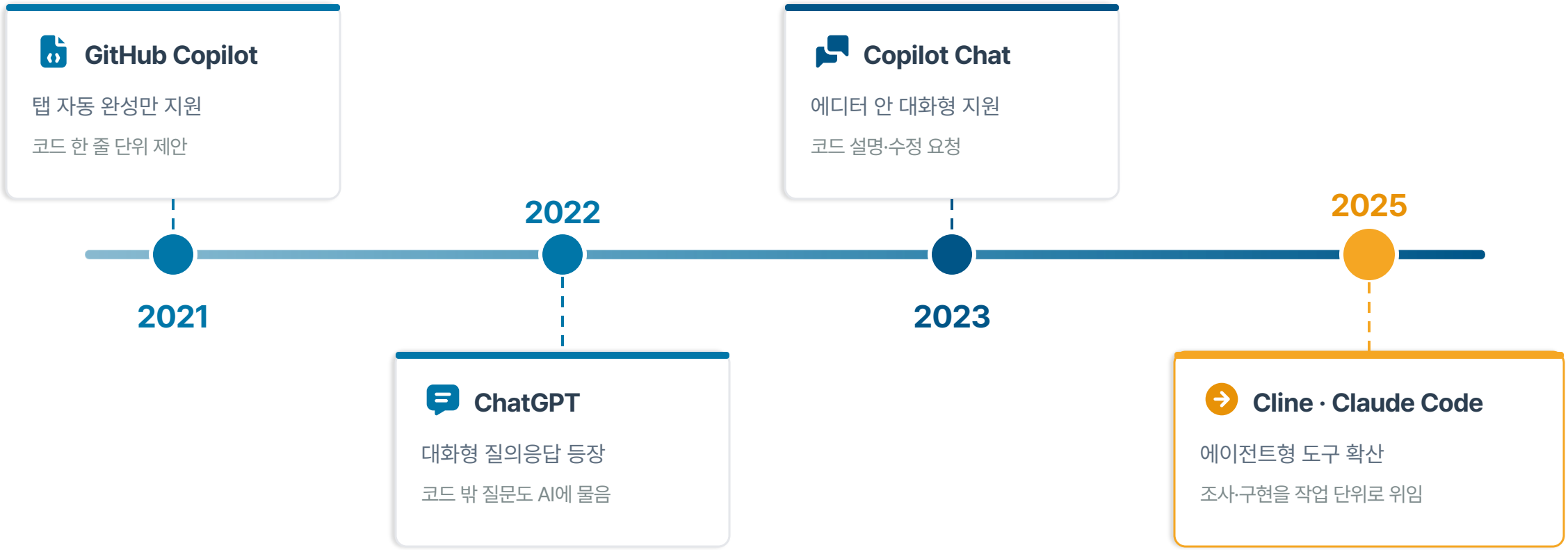
바뀌는 것: 기술(수단)

더 나은 수단으로 패러다임이 전환됨

Source: 원 발표 26쪽, 원 발표가 참고한 LayerX 마쓰모토 유키 발표 Building Products in the LLM Era(원 발표 기준, 미확인)

KEY INSIGHT 뒷부분(요구 전달 능력)은 해설 보충

KEY INSIGHT AI에 맡기는 단위가 코드 한 줄 제안에서 작업 위임으로 커짐 — 사람의 역할이 확인과 판단 쪽으로 옮겨 감



보충: 개발자 84%가 AI 도구를 쓰거나 쓸 계획 (Stack Overflow 2025)

Source: 원 발표(스리세이크, 2026.02.04) 발표자와 AI 연표 쪽; Stack Overflow Developer Survey 2025 (survey.stackoverflow.co/2025/ai)

연도 배치는 원 발표 연표 기준. 카드의 회색 보조 문구와 설문 수치는 보충

KEY INSIGHT 콘텐츠에서 일어난 개인화가 소프트웨어에서도 일어남 — 앱 수가 늘면 검색·유통·보안 검증 방식이 다시 짜임

만드는 주체의 변화

콘텐츠 영역 (이미 일어난 변화)

방송사·출판사 중심
제작 장비와 유통 채널이 필요

→

개인 크리에이터 중심
블로그·영상 플랫폼에서 직접 발행

소프트웨어 영역 (지금 일어나는 변화)

개발 회사·개발팀 중심
개발 인력과 기간이 필요

→

개인·현업 담당자 제작 증가
개발 에이전트로 앱을 직접 만듦

발표 자료의 질문

개인 쪽으로 무게가 옮겨 갈 때 바뀌는 인터넷의 모습

한 사람만 쓰는 앱·소규모 앱이 크게 늘어남

국내 관점 (보충)

- 현업이 만든 앱의 보안 점검 주체와 시점
- 운영·장애 대응 책임을 누가 지는지
- 사내 데이터 접근 권한 부여 기준

Source: 원 발표 14쪽(생각할 질문 — 콘텐츠처럼 소프트웨어도 개인으로). 영역별 설명과 국내 관점은 보충

KEY INSIGHT 프로덕트 개선 주기는 확실히 빨라짐 — 다만 이 속도를 안전하게 유지할 구조는 저절로 생기지 않음

변화 1

개발에 참여하는 사람이 늘어남

엔지니어가 아니던 직군도 프로덕트 개발 가능
AI로 고객에게 직접 가치를 전달함

변화 2

프로덕트 개선 주기가 빨라짐

기능 하나를 내놓기까지 걸리는 시간이
눈에 띄게 짧아짐

발표자의 질문

모든 문제가 해결됐나?

속도는 빨라졌지만
빨라진 속도를 안전하게 유지할
구조는 저절로 생기지 않음

이 구조를 만드는 사람은?

Source: sushichan044, Giiku CAMP 아카데미아 발표(2026-07-08)

KEY INSIGHT 매일 많은 사용자가 쓰는 기존 제품과 복잡한 운영에 어떻게 넣을지가 실제 과제임 — 시제품 속도만으로는 성과가 나지 않음

쉬워진 것

AI로 간단한 프로토타입을 만드는 일

직군과 관계없이 누구나 할 수 있는 수준으로 대중화됨

실무에서 그대로 남은 제약

대규모 제품

매일 많은 사용자가 쓰는 거대한 코드와 화면

복잡한 업무 프로세스

여러 부서가 얽힌 실제 운영 절차

기술 부채

쌓여 온 구조적 제약과 오래된 코드

전체 정합성

기능 하나가 다른 기능과 맞물려야 함

실제 과제

매일 많은 사용자가 쓰는 기존 제품과 복잡한 운영에 AI 개발 방식을 어떻게 넣을 것인가

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 13쪽

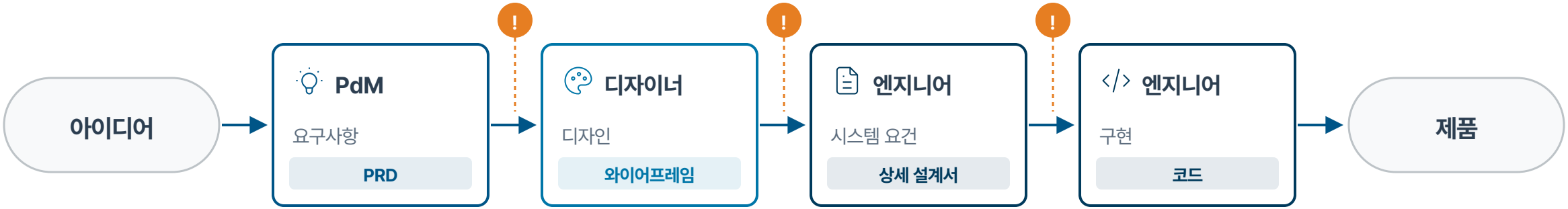
SECTION 3.2

1인 개발과 작은 팀

인계 비용을 없앤 1인 개발과 2~3명 팀

3.2 직군 간 인계 비용

KEY INSIGHT 인계 지점마다 대기와 Context 손실이 생김 — 현장에서 배우는 속도가 그만큼 떨어짐



넘길 때마다 같은 손실이 반복됨

커뮤니케이션 비용

의도를 설명하고 질문에 답하는 회의·문서 작업이 인계마다 생김

대기 시간

앞 직군의 산출물이 나올 때까지 다음 직군이 기다림

Context 일부 손실

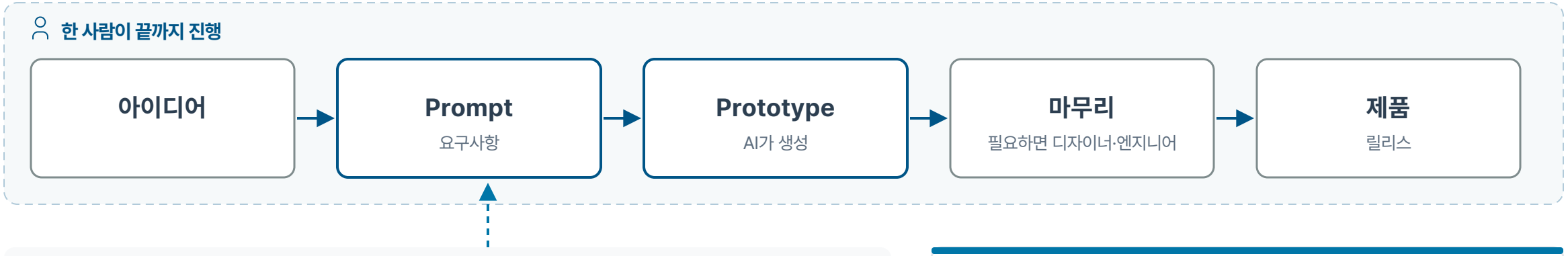
문서로 옮겨 적을수록 왜 만드는지가 빠짐
PRD·디자인·설계서 모두 해당

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 11쪽

PdM(Product Manager, 제품 관리자) · PRD(Product Requirements Document, 제품 요구사항 문서)

3.2 1인 end-to-end 개발 프로세스

KEY INSIGHT 한 사람이 기획부터 릴리스까지 맡으면 인계 비용이 사라짐 — 팀으로 일해도 동작하는 시제품을 두고 논의하면 논의가 구체화됨



팀으로 일할 때도 효과 있음

동작하는 시제품을 빠르게 만들어 두고 논의하면
말로만 하던 논의보다 해상도가 높아짐

없어지는 비용

- 커뮤니케이션 비용 없음
- 인계 대기 시간 없음
- Context 손실 없음

발표 자료도 극단적인 예로 제시

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 12쪽

3.2 AI 시대 팀은 2~3명 크로스펑셔널로 작아짐

KEY INSIGHT 팀 안에서 끝나는 일이 늘어 몹을 하기 쉬운 조건이 갖춰짐 — 작은 팀은 생산성 손실을 덜 걱정해도 됨

변화 1

팀의 소규모화

AI가 일손을 보태면서
2~3명 팀이 적정 규모가 됨



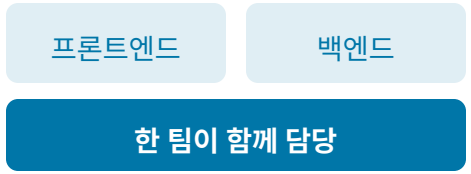
2~3명

변화 2

크로스펑셔널 팀

기술 영역별로 나누지 않음
예: 프론트엔드·백엔드 분담 없음

인접 영역도 AI 지원으로 맡기 쉬워짐



몹을 하기 쉬운 조건

팀 안에서 끝나는 일이 늘어남

- 작은 팀은 생산성 손실을 덜 걱정해도 됨
- 다른 팀을 기다리는 인계가 줄어 전원이 같은 일에 모이기 쉬움

발표 자료자의 전망이며 정량 근거는 제시되지 않음

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07)

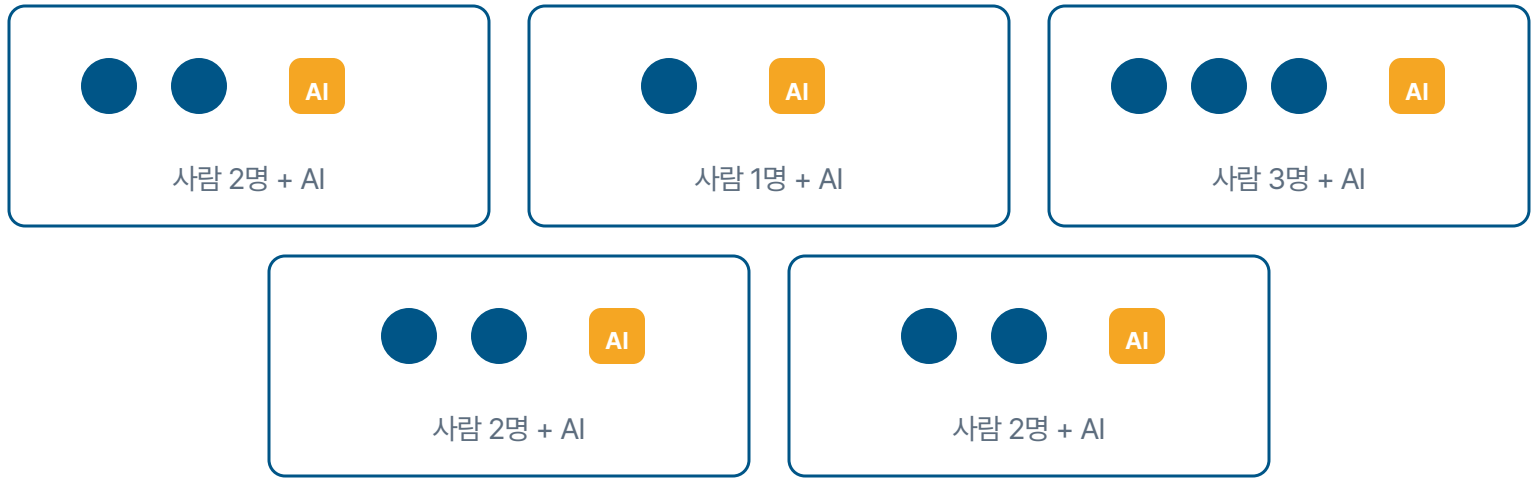
3.2 사람과 AI 혼합 소규모 팀 확대

KEY INSIGHT 팀 크기가 10명 이하에서 1~3명 수준으로 작아짐 — 대신 여러 팀이 함께 일하는 협업 상황이 늘어남

기존: 6명 × 2팀



앞으로: (사람 1~3명 + 생성형 AI) × 5팀



AI가 채우는 부분

다양한 도메인 지식, 프로그래밍, 설계 능력을 갖춤. 사람만 할 수 있는 영역은 여전히 남음

팀 크기

스크럼 가이드 2020의 10명 이하 기준보다 더 작아짐
적은 인원의 경험·기술은 AI가 보완

새 과제

작은 팀이 늘어 여러 팀이 함께 개발하는 협업 상황이 많아짐

Source: 오요베 다카오(HoloLab), Scrum Matsuri 2025 후쿠오카 트랙 발표(2025.10.4) · 스크럼 가이드 2020 — 팀별 인원 배분은 예시

3.2 구현 장벽 하락과 직무 경계 문제

KEY INSIGHT 위임 범위·스펙 파악 수준·품질 책임을 팀이 다시 정해야 함 — 기준이 없으면 품질을 책임질 사람이 비게 됨

일어나는 일

코드 작성 자체는 AI가 상당 부분을 맡을 수 있게 됨

구현 장벽 하락

코드를 쓰는 데 드는 시간과 숙련이 줄어듦

직무 경계가 흐려짐

기획·디자인 담당도 코드를 만들기 시작함

역할이 계속 바뀜

한 번 정한 분담이 오래 유지되지 않음

팀이 답해야 할 질문

어디까지 AI에 맡길지

위임할 작업과 사람이 직접 할 작업의 경계

스펙과 코드를 어디까지 파악할지

AI가 만든 결과를 이해하는 수준의 기준

품질은 누가 어떻게 보장할지

리뷰·테스트 책임자와 방법의 재정의

Source: 원 발표(TRAILBLAZER Kazuhisa Naoi, 2026.08.21) 직무 경계 쪽

세 가지 일과 질문은 원 발표 요지. 각 카드의 회색 설명 줄은 보충

KEY INSIGHT 한 영역만으로는 기술 기여의 총량을 유지하기 어려움 — 학습 비용도 낮아져 영역을 넘기 쉬워짐

출발점

개별 기술 세부 지식의 가치 감소

↓

필요한 것

여러 영역에 기여해 기술 기여의 총량을 유지

↓

학습 비용도 낮아짐

AI에 물어 가며 새 영역을 익히는 속도가 빨라져 적극적으로 영역을 넘을 수 있음

풀스택화 — 기술 영역을 옆으로 넓힘

프런트엔드만 담당	→	**백엔드까지 담당**
애플리케이션만 담당	→	**인프라·ML까지 담당**

풀사이클화 — 공정을 위아래로 넓힘

미들·시니어 엔지니어에게 요구되는 단계

구현에 더해 요구사항·설계 같은 상위 공정과 운영까지 함께 맡음

Source: 원 발표(스리세이크, 2026.02.04) 풀스택이 되기 쪽

풀사이클의 운영 포함 해석과 학습 비용 설명 문구는 보충

SECTION 3.3

소프트웨어 팩토리 사례

freee 의 24시간 개발 공정

3.3 개발·기획 리드타임 60~90% 단축

KEY INSIGHT 하네스 운영 방식만 바꿔도 AI 가동 시간이 68% 줄었음 — 사람 대기 시간을 뺀 값이라 대기 단축은 별도 과제로 남음

개발

하네스 운영 방식 변경 전후 실측 (요건 → 프로토타입, 사람 대기 제외)

AI 가동 시간



-68%

후속 수정 시간



-80%

기획 문서

MRD · 요구 사양서 작성 시간 (2026년 7월, 여러 제품에서 실증)

제품	기존	도입 후	단축률
지출	4시간	1시간	**75%**
고정자산	1주	2~3일	**약 60%**
신고	2일	2시간	**약 88%**
IT 관리	2일	2시간	**약 88%**
인사노무	2일	2시간	**약 88%**

개발과 기획 양쪽 모두 리드타임 60~90% 단축

MRD(Market Requirements Document, 시장 요구 문서)

Source: freee 사내 실측, AI×개발조직 Summit 2026 특별강연 원 발표 20쪽

KEY INSIGHT 모자란 사양·법령 자료는 공정이 보충해 되돌려 줌 — 다음 안건부터 같은 조사가 줄어듦

공정 흐름 — 각 공정의 AI가 하는 일

탐색	기획	설계	구현	릴리스	운영
PdM	PdM	Eng	Eng	QA · PdM · Eng	CS · PdM
기획안을 넓히고 고름	요구 사양 작성	설계 · 테스트 설계	코드 · 테스트 작성	사양과 구현의 차이 검출	문의 · 장애에서 학습

참조 · 부족한 지식은 갱신

지식 베이스 (Knowledge Base)

여러 데이터 저장소를 추상화해 부르는 이름

현행 사양	용어집	법령 · 제도	의사결정 기록	장애 · 결함	고객 지원 이력
고객 요청	QA 지식	백로그	로드맵	사업 전략	시스템 의존 관계

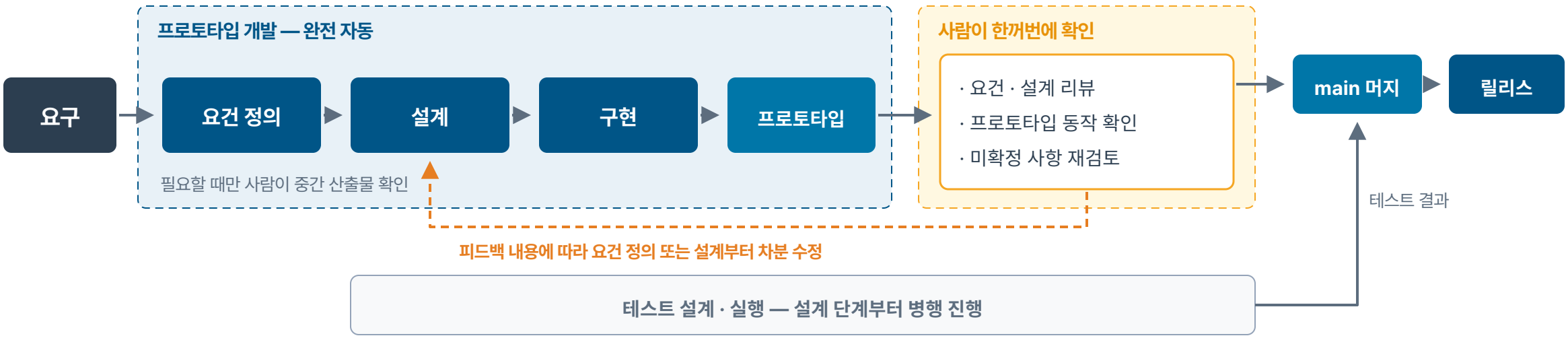
① 가운데에 지식 베이스
이미 아는 지식을 AI가 직접 참조할 수 있는 곳에 둠

② 모든 공정의 AI가 참조
탐색부터 운영까지 같은 곳에서 지식을 가져옴

③ 모자라면 갱신해 되돌림
기존 사양 · 법령 자료가 부족하면 지식 베이스 갱신으로 반영

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 22쪽 (기획과 개발이 공유하는 지식 베이스 개념도)

KEY INSIGHT 사람 확인을 프로토타입 뒤 한 번으로 모아 피드백 주기를 줄임 — AI가 임시로 정한 판단은 미확정 사항으로 남겨 추적함



지식 베이스

기존 사양 · 용어 · 업무 규칙 / 설계 방침 · 과거 의사결정 — 여러 정보원을 가로질러 참조하는 공통 기반

참조 · 갱신
공정마다 목적과 필요한 정보를 참조, main 머지 때 갱신

미확정 사항
AI가 임시로 정한 판단은 따로 남겨 사람이 승인

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 12~13쪽 (시프트 레프트와 피드백 주기 단축)

KEY INSIGHT 계약과 검사를 기계가 먼저 맡아 사람 개입을 줄임 — 자동 수정에는 상한을 두고 수렴하지 않으면 사람에게 넘김

Orchestrator 의존 순서대로 실행 · 완료 여부를 기계로 검사 · 중단 지점에서 재개 · 시간 · 비용 · 사람 개입 기록

가이드 — 맥락과 작업 규칙

· 용어집 · ADR · 기존 코드 조사 결과를 작업 전에 반드시 공유

· 출력 형식 · 단위 · 공정 간 인계 방식을 계약(Contract)으로 고정

센서 — 기계 검증

· 요건과 테스트의 대응을 대조해 누락을 나열

· 필수 산출물과 계약을 검사하고, 모두 충족해야 다음 공정으로 진행

리뷰 — AI와 사람의 이중 확인

· 관점별로, 근거를 붙여 지적. 자동 수정은 상한을 두고 수렴하지 않으면 사람에게 넘김

· 사람은 전문을 읽지 않고 판단할 요점만 확인

후속 수정 — 상류부터 일괄 수정

· AI의 임시 결정과 사람의 승인을 구분해 기록 · 추적

· 오류가 들어간 공정을 찾아 상류부터 일괄 수정하고 해당 게이트로 재진입

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 14쪽 · [보충] 가이드/센서 구분은 Thoughtworks의 Harness Engineering 개념(2026)과 같은 틀

밤에는 AI가 작업하고 아침에 사람이 판단함

KEY INSIGHT 사람의 확인 범위를 판단점으로 좁혀야 AI 속도를 살릴 수 있음 — 이를 위해 클라우드 환경·미확정 사항 목록·리뷰 View를 갖춤

밤

사람이 지시를 넣음

→

AI 가동

요건 → 설계 → 구현 → 테스트

테스트는 병렬 진행

→

아침

판단점만 확인

→

차분 재실행

요건 · 설계 단계를
기점으로 일괄 수정

① 클라우드 개발 환경

- 개인 단말에 의존하지 않고
  사람이 없는 동안에도 처리 계속
- 진행 상황과 산출물을 팀이
  공유하고 중간에 판단에 참여

② 미확정 사항 목록

- AI가 임시로 정한 판단을
  미확정 사항으로 남김
- 사람의 승인과 구분해 기록 · 추적
- 아침에는 임시 결정 승인과
  남은 쟁점만 다룸

③ 사람용 리뷰 View

- 설계서는 AI 용으로 줄이지 않고
  망라해 남김
- 사람은 전체 그림 · 사실 · 이유 ·
  질문만 담은 별도 View로
  판단할 곳만 봄

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 15쪽

KEY INSIGHT Factory Owner는 출력 품질을, Outcome Owner는 고객 가치를 책임짐 — 무엇을 AI에 위임할지는 이 두 사람이 정함

운영 조직 — Software Factory를 기반으로 세 조직이 역할을 나눔

제품팀 Eng · PdM · QA

Factory를 실제로 사용

· 실제 안건을 Factory에 투입
· 사람이 판단한 관점을 지식으로 환원
· Factory Owner · Outcome Owner 담당

초고속 기획 프로젝트

Factory 입구를 정비

· 기획 · 개발 공용 지식 베이스 정비
· PdM이 정할 요구 범위와 설계 ready 게이트 정의
· 여러 제품에서 실증 (2026년 7월~)

초고속 개발 랩 CTO실

Factory 내부를 개발

· 하네스와 모듈화된 파이프라인 개발
· 평가 데이터셋과 Factory 계측 기반
· 개선한 체계를 각 팀에 배포

사람의 역할 — 품질과 고객 가치에 책임지고, 실행은 AI가 맡음

Factory Owner 출력 품질 유지 — 라인과 검증을 설계

· 검증 설계를 충분히 두껍게 유지
· 무엇을 합의하고 무엇을 맡길지 결정
· 실행 환경과 권한 정비
· 기준을 벗어나면 라인을 정지

Outcome Owner 고객 가치 전달 — 요구 투입과 출시 판단

· 풀어야 할 과제와 우선순위 결정
· 수용 기준(Acceptance Criteria) 정의
· 출시 여부 결정
· 왜 이 방침인지 기록으로 남김

Source: freee, AI×개발조직 Summit 2026 특별강연 원 발표 17~18쪽

SECTION 3.4

사람이 맡는 확인 지점

머지·배포·리뷰처럼 사람이 남는 단계

KEY INSIGHT 에이전트 작업은 커밋·Push·CI 통과까지로 한정됨 — 머지·배포와 중요한 의사결정을 사람에게 남겨 사고 범위를 제한함

1 티켓은 반드시 한 건씩 처리

AI 수행

도중에 다른 티켓 착수 금지

2 1 커밋 = 1 티켓

AI 수행

커밋 메시지에 티켓 정보 포함

3 처리 방법이 명확한 티켓만 진행

AI 수행

명세 확인이 필요하면 질문을 적고 건너뜀

4 중요한 의사결정은 PM 답변 대기

사람 관여

명세·데이터 모델 변경과 파괴적 변경은 티켓에 기록

5·7 구현 뒤 테스트, 커밋 전 빌드 점검

AI 수행

lint·타입·spec·헬스 체크 통과를 로컬에서 확인

8 Push 뒤 CI 통과까지 감시

AI 수행

실패하면 코드를 고쳐 다시 Push

9 머지·배포는 사람 책임

사람 관여

AI는 커밋·Push·CI 통과까지만 담당

11 새 전제 지식은 하네스 갱신 제안

사람 관여

Skills·CLAUDE.md 수정안을 사용자에게 보고

Source: 원 발표 13쪽, 하네스용 SKILL.md 대원칙 12개 중 8개를 요약 (번호는 원문 순서)

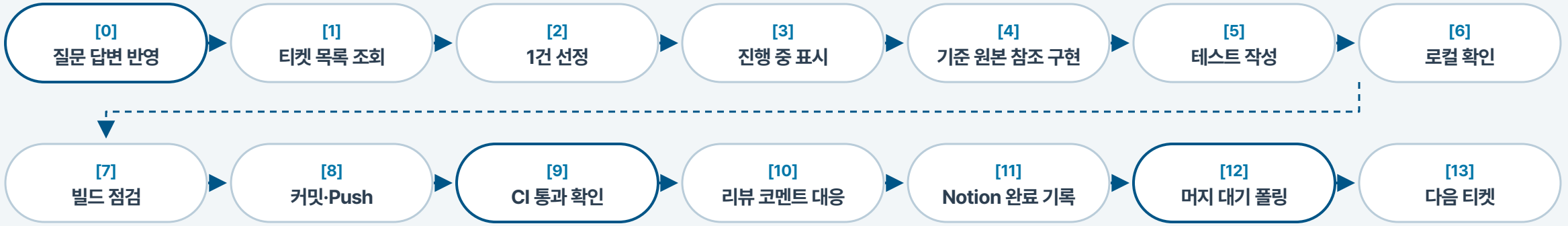
6번(Notion 기록 범위 제한)·10번(머지 대기 폴링)·12번(자기 PR 리뷰 코멘트 자동 대응)은 다음 장 개발 루프에 반영

3.4 AI·사람 책임 범위별 개발 루프

사람이 맡는 확인 지점

KEY INSIGHT 사람 개입 지점이 답변·리뷰·머지·배포 넷으로 고정됨 — 병목이 어디서 생기는지 바로 드러나 운영 부담을 예측 가능

AI 책임 범위 · 14단계 반복



사람 책임 범위 · 4단계

H1	질문 티켓에 답변 기입	다음 루프의 [0] 에서 반영
H2	PR 리뷰·변경 요청	[10] 에서 AI가 대응
H3	PR 머지	[12] 가 머지 여부를 폴링
H4	배포	머지 뒤 사람이 실행

▶ 실행: Claude Code의 **/loop** 와 스킬 명령으로 루프를 계속 반복

계속 돌게 하는 요령: 착수할 티켓이 없으면 10분 기다린 뒤 다시 확인하도록 지시

Source: 원 발표 14~16쪽 (루프 정의·실행 방법·루프 구조도) · code.claude.com/docs (Claude Code /loop·skills)

단계 이름은 원 발표 구조도를 한국어로 다시 적은 것. /loop 동작 세부는 Claude Code 버전에 따라 다를 수 있음(보충)

3.4 개발 공정별 AI와 사람의 역할 분담

사람이 맡는 확인 지점

KEY INSIGHT 초안·조사·1차 리뷰는 맡기고 결정·완성·최종 확인은 직접 함 — 단계마다 누가 확인하는지 팀 규칙으로 정해야 품질이 유지됨

	요구사항 정의	설계	구현	리뷰
AI 이전 전부 사람 손	PO 인터뷰 진행 요구사항 결정	기존 구현을 직접 확인하며 설계·작업 분할·견적	여러 번 검색하며 직접 구현, 셀프 리뷰	PR 을 내고 리뷰 요청, 경우에 따라 여러 번 왕복
AI 이후 전 공정에 AI	PO가 AI와 미리 정리한 내용으로 인터뷰 **사람: 요구사항 결정**	AI가 기존 구현을 조사, AI와 상의하며 설계 작업 분할·견적도 AI **사람: 설계 판단**	AI가 80% 작성, 조사·셀프 리뷰도 AI **사람: 나머지 완성**	AI 리뷰를 먼저 통과한 뒤 사람 리뷰 진행 **사람: 최종 리뷰**

Source: 원 발표(스리세이크, 2026.02.04) 생성형 AI 이전·이후 개발 업무 전체 쪽

구현 80%는 발표자 체감 비율. 노란 칸의 사람 역할 요약은 보충

3.4 AI-DLC 개발 방식의 역할 분담

사람이 맡는 확인 지점

KEY INSIGHT

AI가 작업 계획과 분해를 주도하고 사람은 검증·승인을 맡음

결과물 품질은 사람의 판단 수준에 달림

AI-DLC(AI-Driven Development Life Cycle, AI 주도 개발 수명주기): AWS가 2025년에 제안한 개발 방법론

기존 방식

작업 실행

사람이 작업을 직접 실행

AI 역할

일부 작업을 돕는 지원 도구

AI-DLC

AI 역할

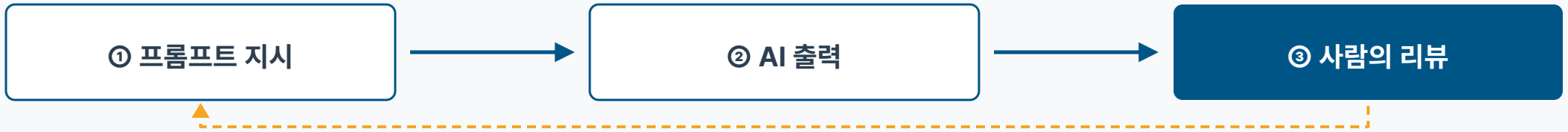
작업 계획 작성과 작업 분해를 주도

사람 역할(PdM · 디자이너 · 엔지니어)

AI 결과를 검증하고 승인 · 의사결정

단계마다 반복하는 주기

주기를 빠르게 돌려 시제품을 만듦



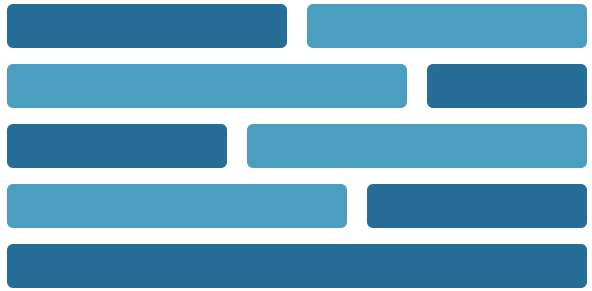
Source: 원 발표(2026-06-18); AWS DevOps Blog, AI-Driven Development Life Cycle (aws.amazon.com/blogs/devops/ai-driven-development-life-cycle)

3.4 AI 개발의 병목은 사람의 리뷰 속도

KEY INSIGHT AI를 더 써도 결과물은 늘지 않음 — 확인 범위를 줄이지 못하면 AI 속도가 낭비됨

AI 코드 생성

짧은 시간에 대량의 프로그램 생성



처리 속도와 품질이 계속 올라감

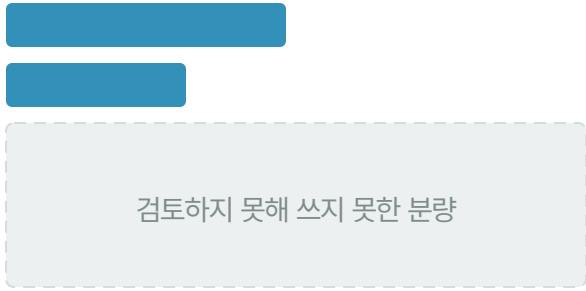
사람의 코드 리뷰

인지 부담을 넘어선 분량

다 소화하지 못함

결과물

사람의 처리 능력만큼만 나옴



AI가 발전해도 처리량이 그대로임

발표 요지

· AI의 처리 속도와 품질이 사람의 인지 부담을 크게 넘어섬

· 사람이 리뷰를 다 하지 못해 AI를 다 쓰지 못하는 상태 — 사람이 병목이 됨

해법 방향: 사람이 어디까지 확인할지 먼저 정해 확인 범위를 줄임

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 8쪽

KEY INSIGHT 반도체·OS·라이브러리도 내부를 다 모른 채 써 왔음 — AI가 쓴 코드도 계층별 테스트로 품질을 보증하면 됨

기술이 발전할 때마다 일정한 블랙박스화를 받아들여 온 흐름



발표 요지

· 각 계층은 블랙박스화됐지만 자기 책임 범위에서 품질을 보증해 왔음

· AI 소프트웨어 개발도 같은 방식의 품질 보증이 필요함 — 테스트로 품질을 지키되 내부를 다 읽지 않는 것은 받아들임

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 10쪽

SECTION 3.5

하네스와 컨텍스트 설계

규칙 파일·스킬·컨텍스트를 팀이 함께 관리

KEY INSIGHT 규칙·스킬·메모리·피드백이 함께 있어야 출력이 일정해짐 — 하나라도 빠지면 같은 실수를 사람이 반복해서 바로잡아야 함

에이전트 하네스: AI 에이전트가 예측 가능하고 품질 높게 동작하도록 만드는 제어 계층

규칙 파일·스킬 정의·메모리·피드백 루프를 하나의 개발 환경으로 묶어 설계

규칙
지켜야 할 기준

- CLAUDE.md
- .cursorrules
- 코딩 규약

스킬
반복 작업 절차·도구

- SKILL.md
- MCP 서버
- 사용자 정의 도구

메모리
참고할 지식

- 과거 판단 이력
- 프로젝트 지식
- 명세서 참조

피드백
결과 검증 신호

- Lint·테스트 결과
- 리뷰 지적
- CI/CD 결과

규칙이 기준을, 스킬이 절차를, 메모리가 맥락을 주고 **피드백이 다음 실행의 규칙을 고침**

한 요소만 갖춘 상태에서는 같은 지적을 사람이 매번 다시 해야 함

Source: 원캐리어 AI 프로덕트 개발실장, Findy AI+ Harness Engineering Meetup (2026.05.22)

구성 요소 분류와 예시 파일은 원 발표 기준. 원 발표는 스킬 파일을 SKILLS.md 로도 적었으나 Claude Code 공식 이름은 SKILL.md

KEY INSIGHT 규칙을 파일로 남기면 팀 전체가 같은 지시를 재사용함 — 개인 노하우에 머물던 작업 방식을 조직 단위로 확장 가능

프롬프트 엔지니어링

한 번의 대화를 다듬는 일

대상	한 번의 대화를 최적화
작성 방식	매번 프롬프트를 새로 작성
수명	그 자리에서만 쓰는 지시
공유	개인 노하우에 머묾
확장	사람 수만큼 품질이 갈림

하네스 엔지니어링

하네스를 설계·투입하고 피드백으로 계속 다듬는 일

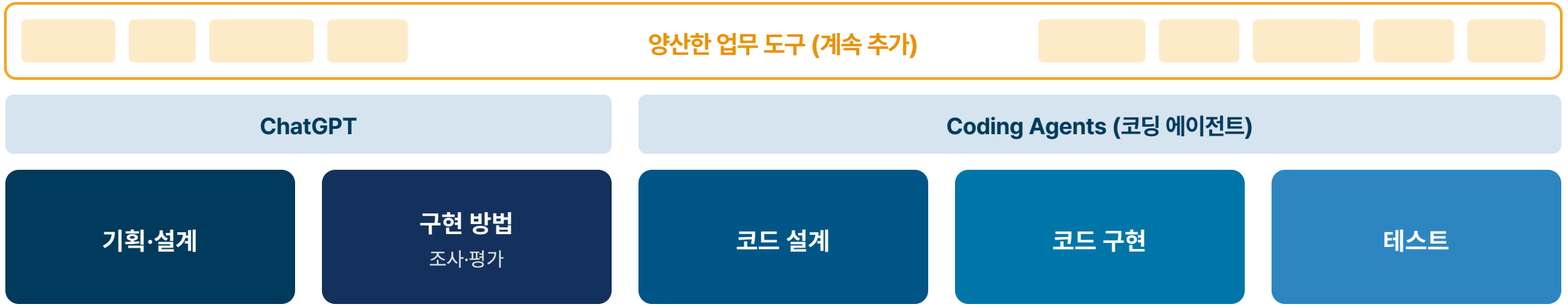
대상	지속되는 개발 환경을 설계
작성 방식	규칙·스킬을 파일로 정의
수명	저장소에 남아 계속 재사용
공유	팀 전체가 같은 파일을 사용
확장	**피드백으로 개선하며 조직 단위로 확장**

Source: 원 발표 10쪽 · Anthropic Engineering, Harness design for long-running application development (anthropic.com/engineering/harness-design-long-running-apps)

비교 항목은 원 발표 기준

KEY INSIGHT 범용 AI 도구 위에 팀 전용 도구 층이 더해짐 — 팀의 반복 업무에 맞춘 개선이 N 을 키우는 다음 수단이 됨

3층 구성: 팀 전용 도구 · 범용 AI 도구 · 개발 업무 5단계



팀 전용 도구 층이 범용 도구가 닿지 않는 작업을 맡음

배포·QA 전달·이벤트 목록처럼 팀마다 다른 반복 작업은 범용 도구만으로 처리하기 어려움

발표 자료는 이런 도구를 계속 만들어 효율 배수 N 을 키우는 것을 다음 단계로 제시

N → N+

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 9쪽

3층 도식은 원 발표 기준. 하단 설명의 반복 작업 예시는 6쪽 도구 목록에서 가져온 보충

KEY INSIGHT 구현 비용이 낮아져 우선순위에서 밀리던 내부 도구가 실제로 만들어짐 — 팀의 반복 작업을 줄이는 수단이 늘어남

아이디어를 구현하는 비용이 줄어 그동안 미뤄 두던 팀 내부 도구를 직접 만듦

개발·배포 3종

배포 자동화
배포 절차를 스크립트로 묶음

코드 분석 도구
코드베이스 구조·현황 파악

Pixel 차이 검출 도구
디자인 시안과 화면 차이 확인

품질·QA 2종

QA 용 앱 배포 도구
검증용 빌드를 QA에 전달

이벤트 트래킹 목록 도구
수집 이벤트 정의를 목록화

지식·관측 2종

사내 Wiki MCP 서버
에이전트가 사내 문서 조회

Coding Agent 사용량 그래프
팀의 에이전트 사용량 시각화

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 6쪽

도구 이름 7종은 원 발표 기준. 세 묶음 분류와 도구별 한 줄 설명은 이해를 돕기 위한 보충

KEY INSIGHT 에이전트가 사내 규칙과 문서를 직접 참조함 — 팀 고유 맥락을 매번 프롬프트에 붙여 넣는 수고가 줄어듦

연결 구조



얻는 것

- 에이전트가 팀 고유 규칙·문서를 직접 조회
- 사내 맥락을 반영한 답변과 코드 초안
- 프롬프트에 문서를 붙여 넣는 수고 감소

MCP 란(보충)

- Model Context Protocol
- 2024.11 Anthropic이 공개한 개방형 표준
- AI 도구와 외부 데이터 연결 규약

함께 정할 것(보충)

- 문서별 접근 권한 범위
- 조회 기록 보관
- 대외비 문서 노출 차단

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 6쪽; anthropic.com/news/model-context-protocol(2024.11.25)

사내 Wiki MCP 서버는 원 발표의 도구 목록 기준. 연결 구조·MCP 설명·권한 항목은 보충

💡 **KEY INSIGHT** 모델과 도구는 IT가 고르지만 맥락과 업무 규칙은 업무 전문가만 정확히 씀 — 에이전트 품질의 책임은 업무 맥락을 아는 사람에게 있음

계층	담는 내용	책임 주체
모델	기반 모델의 선택과 버전	공급사·IT 부서
도구	검색, 사내 시스템 연동, 파일 처리 기능	IT 부서
맥락	목적, 처리 대상, 참조 자료, 예시	업무 전문가
업무 규칙	판정 기준, 예외 처리, 승인 조건	업무 전문가

맥락 설계 원칙: 목표에 필요한 정보 가운데 가장 적고 신호가 강한 묶음을 넣음 — 긴 작업은 요약 압축·작업 메모·하위 에이전트 분리로 맥락을 관리

SECTION 3.6

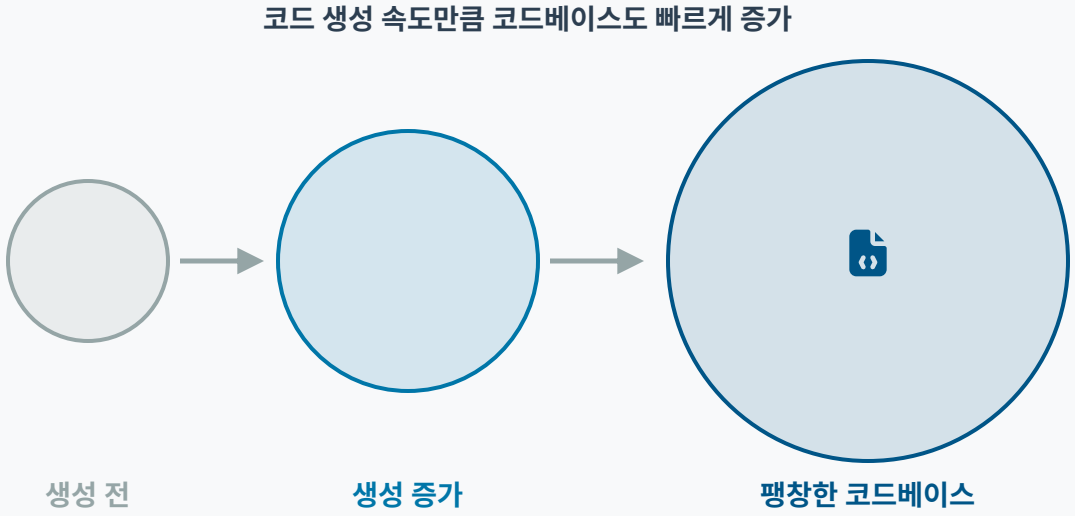
속도가 만드는 설계·운영 부담

코드 총량·기술 부채·운영 책임

3.6 코드가 불어날수록 설계 붕괴도 빨라짐

KEY INSIGHT AI가 만든 코드도 사람이 구조를 관리해야 함 — 방치하면 수정 범위가 넓어져 AI 작업 품질도 떨어짐

코드 생성량이 늘면 코드베이스가 불어나고 설계 붕괴도 함께 빨라짐



설계 붕괴가 나타나는 모습(보충 예시)

중복 코드
비슷한 처리가 여러 곳에 생성됨

경계 없는 의존
모듈 사이 호출이 얽혀 수정 범위가 커짐

일관성 없는 구조
요청마다 다른 방식으로 구현됨

발표 요지 코드 안을 보지 않아도 되는 것은 아님 — 코드베이스가 커질수록 코드 관리는 계속 필요함

Source: 원 발표 자료(Yuki Sensui, LY Corporation, Creators Vision vol.2) 10쪽

팽창 도식과 하단 요지는 원 발표 기준. 설계 붕괴의 세 가지 모습은 보충 예시

AI로 구현이 빨라진 만큼 기술 부채도 늘어남

KEY INSIGHT AI가 혼란에 빠지거나 기존 코드를 깨뜨리는 일도 생김 — 이대로는 AI로 개발을 오래 이어 가기 어려움

확산

AI 에이전트·코딩 특화 모델 등장

AI 주도 개발이 실무에 본격 확산

"AI가 아주 빠르게 구현해 줌"

SNS에서 오가는 체감 사례

"프로덕션 코드 중 xx%는 AI 작성"

기업 발표·뉴스로 나오는 비율 보도

관심은 주로 구현 속도에 쏠림
구조 품질은 뒤로 밀림

새로 드러난 문제

1. **빠른 구현과 함께 기술 부채가 급증**
   구조 검토 없이 늘어난 코드가 다음 변경을 막음
2. **AI가 혼란에 빠져 구현이 끝나지 않음**
   얽힌 코드에서 수정과 되돌림을 반복
3. **AI가 기존 코드와 환경을 깨뜨림**
   바꾸지 말아야 할 곳까지 변경

AI로 개발을 계속 이어 가기 어려움

AI를 써도 개발 규모를 키울 수 없는 상태

Source: 원 발표 자료(MinoDriven, Architecture Conference 2025) 9~10쪽

'AI 작성 코드 비율' 보도는 원 발표가 예로 든 것이며 구체 수치는 원 발표에도 없음

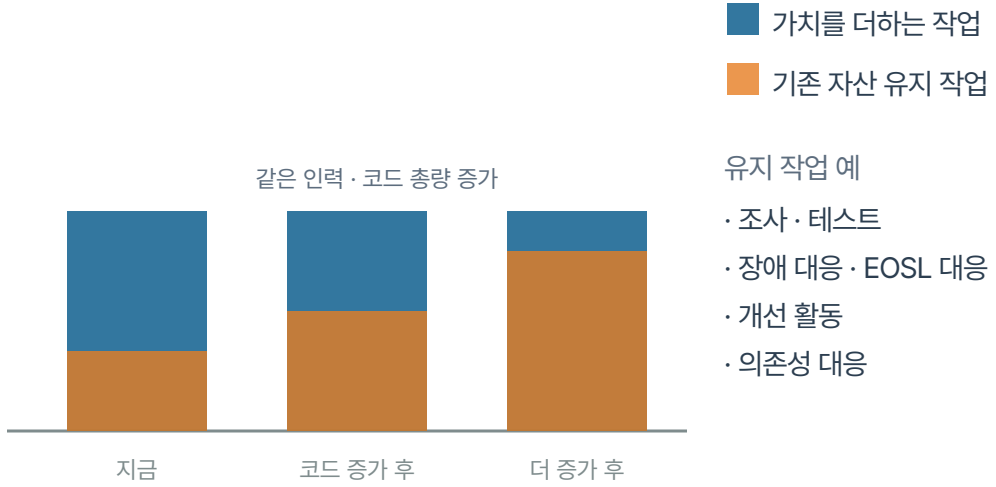
KEY INSIGHT 인력을 고정하면 생산성이 떨어지고 생산성을 유지하려면 조직이 커짐 — 코드는 쓰는 순간부터 유지보수 대상이 되기 때문

변경 한 건마다 따라오는 일

- 기존 코드의 사양 확인
- 영향 범위 조사
- 설계
- 테스트
- 운영 확인

코드가 가치를 내든 내지 않든 같은 비용 발생

개발 작업의 구성 변화



코드 총량이 늘 때의 두 결과

인력을 고정 → 생산성이 계속 떨어짐

생산성을 고정 → 조직이 계속 커짐

Source: 구로다(Kuroda, Indeed Recruit Technologies), 비즈니스 가치와 엔지니어링 2026년도판 발표 자료

오른쪽 막대 비율은 개념도 · 수치 없음

KEY INSIGHT 만들 수 있는 양이 늘수록 계속 운영하는 역량이 가치의 상한이 됨 — 가치를 멈추지 않는 일도 새 기능 개발과 같은 사업 기여임

AI로 만드는 힘 증가
개발 속도 상승 · 비용 하락

→

개발 대상 증가
기능 · 서비스 · 코드 총량

→

유지보수 대상 증가
진짜 병목이 옮겨 가는 자리

함께 늘어나는 유지보수 대상 일곱 가지

- 모니터링
- 장애 대응
- 보안 대응
- 의존성 관리
- EOSL 대응
- 문의 대응
- 자산 제거

가치를 만드는 엔지니어링

새 기능 개발 · 개선 출시

가치 면적의 시작점과 높이를 키움

+

가치를 멈추지 않는 엔지니어링

운영 · 장애 대응 · 의존성 관리 · EOSL 대응

발표 요지: 둘은 같은 사업 기여

Source: 구로다(Kuroda, Indeed Recruit Technologies), 비즈니스 가치와 엔지니어링 2026년도판 발표 자료

AI로 만든 시스템의 운영 책임

KEY INSIGHT 만든 사람이 떠나도 조직이 설명하고 유지보수할 수 있어야 함 — 개발 속도만으로는 이 조건을 채우지 못함

AI로 단숨에 만든 결과물 — 출시 뒤에 남는 질문 10개

01
소유자
시스템의 오너는 누구인가

02
장애 인지
장애를 누가 알아차리는가

03
복구
어떻게 복구하는가

04
유지보수
유지보수 책임은 누구인가

05
인력 변동
담당자가 이동·퇴사하면?

06
블랙박스
내부를 설명할 수 있는가

07
운영 문서
문서가 남아 있는가

08
재현
같은 구성을 다시 만들 수 있는가

09
권한
권한을 회수·정지할 수 있는가

10
종료·이전
종료하거나 옮길 수 있는가

판단 기준: 만든 본인이 없어도 조직으로서 설명·유지보수할 수 있는가

Source: 원 발표 자료 15쪽

질문 10개는 원 발표 도식의 항목을 한국어로 다시 정리함

KEY INSIGHT 그럴듯해 보이는 결과와 올바른 결과는 다름 — 결과를 확인할 책임은 사람에게 남음

01

목업 데이터로 회귀

실제 데이터에 연결했는데
어느새 더미 데이터로 돌아가 있음

02

이른 완료 선언

작업 완료를 보고하지만
실제로는 동작하지 않음

03

문서를 근거로 단정

동작을 확인하지 않고 "문서에
이렇게 적혀 있음"을 사실로 취급

04

구현 누락·회귀

일부 구현을 빠뜨리거나, 고친 곳을
다른 변경으로 다시 망가뜨림

05

승인하지 않은 변경

요청하지 않은 부분까지
임의로 수정

06

가짜 테스트

테스트 통과만을 위해
검증 내용이 없는 테스트 작성

Source: 원 발표 p.11 — 발표자가 AI 에이전트와 실제 작업하며 반복해서 겪은 유형

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

SECTION 3.7

가치 기준의 개발 생산성

출시 속도보다 비즈니스 가치 면적

KEY INSIGHT AI로 만드는 일이 빠르고 싸질수록 무엇을 만들지 묻는 일이 더 중요해짐 — 쓰이지 않는 기능도 같은 속도로 늘기 때문

코드를 더 쓴다고 비즈니스 가치가 그만큼 늘지 않음

코드와 비즈니스 가치는 어떻게 이어지는가

AI로 만드는 일이 빠르고 싸질수록
오른쪽 다섯 질문의 중요도가 올라감

쓰이지 않는 기능도 같은 속도로 늘어남

발표 자료가 던지는 다섯 질문

1. 무엇이 비즈니스 가치에 의미가 있는가
2. 무엇이 비즈니스 가치에 의미가 없는가
3. 무엇을 빨리 만들어야 하는가
4. 무엇을 만들지 말아야 하는가
5. 만든 것을 어떻게 가치로 계속 유지하는가

Source: 구로다(Kuroda, Indeed Recruit Technologies), 비즈니스 가치와 엔지니어링 2026년도판 발표 자료

KEY INSIGHT 출시가 잦아도 쓰이지 않는 기능은 가치를 만들지 못함 — 생산성의 분자를 개발량으로 두면 투자 판단이 틀어짐

사례 1 · 새 기능을 빠르게 계속 출시

겉으로는 빠르고 바쁜 팀



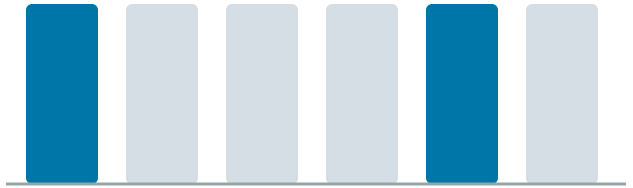
출시한 기능

출시 빈도 높음

생산성이 높아 보임

사례 2 · 같은 속도, 대부분의 기능이 쓰이지 않음

출시 수는 같아도 가치는 거의 없음



회색: 쓰이지 않는 기능

생산성이 높다고 볼 수 없음

가성비 · 시간 대비 효율 · ROI · 생산성의 공통 구조

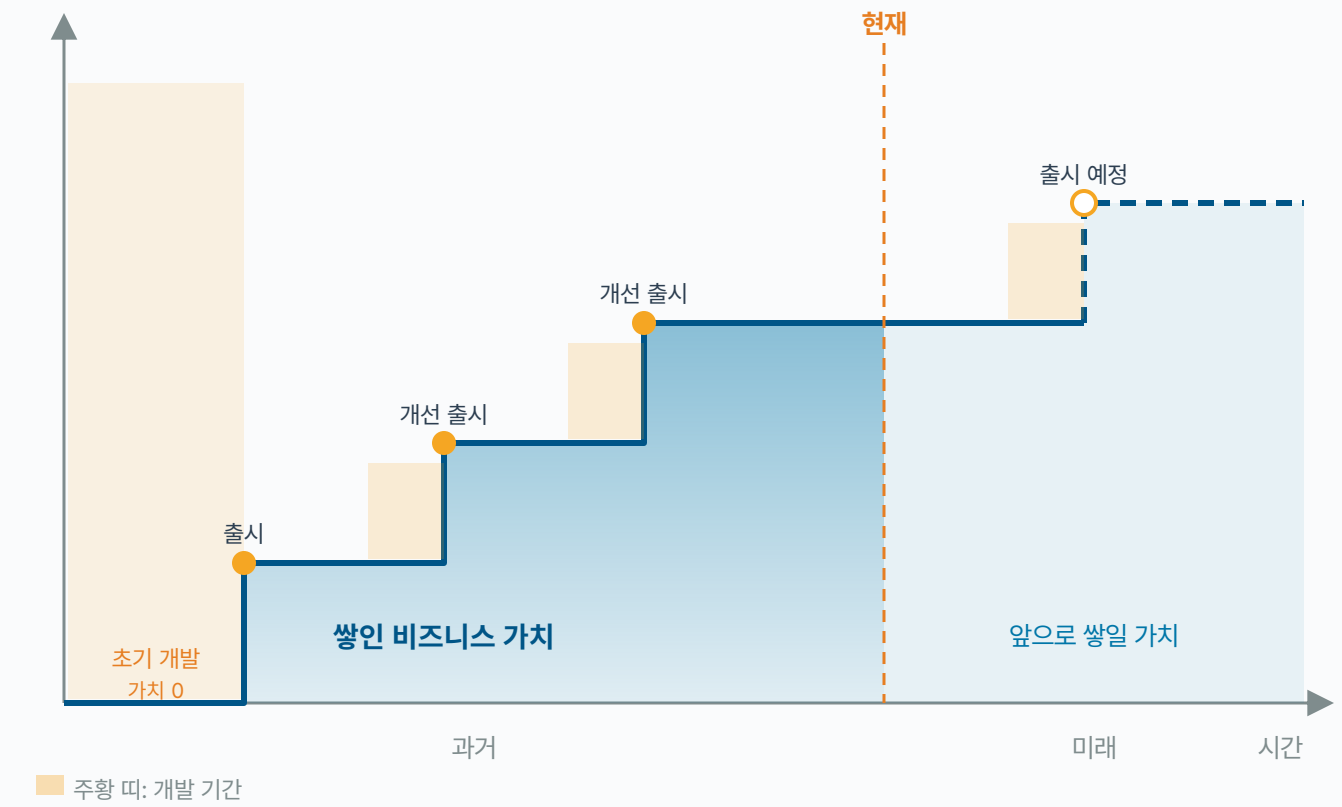
$$\frac{\text{퍼포먼스(분자)}}{\text{비용 · 시간 · 투자(분모)}}$$

분모는 비교적 분명함, 분자인 퍼포먼스는 흐려지기 쉬움

분자를 개발량으로 잡으면 사례 2도 생산성이 높은 팀으로 보임

Source: 구로다(Kuroda, Indeed Recruit Technologies), 비즈니스 가치와 엔지니어링 2026년도판 발표 자료

KEY INSIGHT 개발 중에는 KPI가 오르지 않고 출시 뒤부터 누적됨 — 개발량을 늘려도 출시가 늦어지면 누적 성과가 줄어듦



면적으로 보는 이유

기능·UX 자체는 비즈니스 가치가 아님
출시되고, 쓰이고, 의미가 있을 때
비로소 가치로 전환

키울 대상

개발량보다 시간축 위에 쌓이는
가치의 면적(누적 성과)

AI 시대에 더 중요한 이유

쓰이지 않는 것, 팔리지 않는 것,
운영할 수 없는 것도 빨리 만들 수 있음

과거에 개발해 출시한 기능은
현재 가치의 바닥을 받치는 자산

Source: 구로다(Kuroda, Indeed Recruit Technologies), 비즈니스 가치와 엔지니어링 2026년도판 발표 자료

그래프는 원 발표 도식을 다시 그린 개념도, 수치 없음

3.7 시제품 비용이 회의 비용보다 낮아짐

KEY INSIGHT 먼저 만들어 검증하고 안 되면 버리는 방식이 더 경제적임 — 검토 회의 중심의 개발 절차를 다시 설계할 시점

기존 방식

논의를 끝낸 뒤에 구현

- 아이디어 검토
- 요건 정리
- **기획 회의**
- **설계 리뷰**
- 구현

회의가 늘수록 첫 동작 확인이 늦어짐

생성형 AI 이후

먼저 만들고 결과로 판단

- AI와 아이디어 검토
- AI에 구현 지시
- 바로 실행
- **검증**
- 안 되면 폐기

논의보다 실험·구현이 더 싸고 빠름

인용 요지 — Simon Willison

운영 수준 기능을 구현하는
토큰 비용이, 그 기능을 만들지
말지 논의하는 회의 비용보다
낮아짐

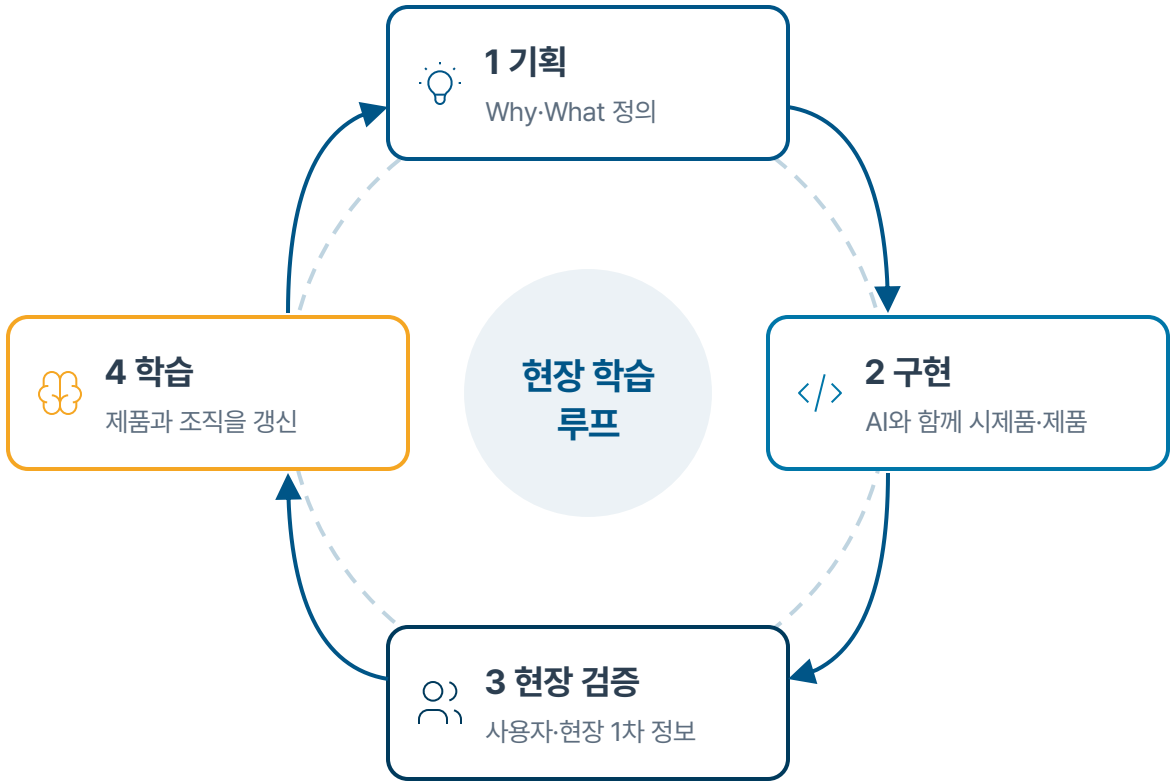
30분 기획 회의를 여는 것보다
실제로 만들어 동작을 확인하는
편이 비용이 적게 듦

전제: 만든 결과를 검증하고
버릴 수 있는 절차가 있어야 함

Source: 원 발표 p.6 · Simon Willison's Weblog(simonwillison.net) 인용

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

KEY INSIGHT 역할을 넘나드는 현상은 루프가 빨라진 결과임 — 조직이 할 일은 이 루프가 멈추지 않게 설계하는 것



AI가 바꾼 것

기획·구현·검증·학습 네 단계 전부가 빨라짐

회전수가 늘수록 현장에서 배우는 양이 늘어남

역할 넘나들기의 위치

루프가 돌기 시작한 뒤 나타난 결과

PdM이 만들고 엔지니어가 기획하는 현상은 루프가 빨라진 뒤에 따라옴

조직이 맡을 일

루프를 멈추지 않게 하는 설계

인계·승인·대기가 루프를 끊는 지점을 찾아 줄임

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 9쪽

무엇을 만들지 정한 뒤 AI로 한 번에 구현

KEY INSIGHT 만들 대상이 명확해지면 구현은 빨라짐 — 병목이 구현에서 기획 판단으로 옮겨 감

STEP 01	STEP 02	STEP 03	STEP 04
고객 요구 파악	**만들 대상 결정**	**AI로 한 번에 구현**	**협력해 출시 완료**
고객 요구 분석 해결할 문제 발견	**왜 만드는지** **무엇을 만드는지** **누구를 위해 만드는지** 사람이 판단하는 핵심 단계	대상이 명확해지면 AI로 빠르게 구현	사내 여러 부서와 협력 출시까지 끝까지 완료

참고 자료

LayerX 사내 자료 "기능을 만들지 마라. 편하게 만들지 마라." — 기능 개발 전에 이유와 대상을 먼저 묻는 원칙

Source: sushichan044, Giiku CAMP 아카데미아 발표(2026-07-08) / 원 발표가 인용한 LayerX 사내 자료

SECTION 04

조직이 바뀌어야 할 것

AI 네이티브 조직·확산·거버넌스·인력 판단·리더·채용·육성·새 직무·학습 조직·손익

SECTION 4.1

AI 네이티브 조직의 조건

도구 보급보다 업무 재설계 역량

4.1 AI 네이티브 조직의 판단 기준은 역량

KEY INSIGHT AI 개입 가능성을 전제로 업무·의사결정을 계속 재설계하는 능력이 핵심임 — 도구 도입·데이터 정비는 그 상태의 도달 조건 일부에 그침

AI 네이티브 조직의 정의 (발표 자료)

**업무 프로세스·의사결정·평가 기준이 AI 개입 가능성을 전제로 설계되고,
그 설계를 조직이 계속 재검토·재최적화하며 바꿔 나가는 능력을 갖춘 상태**

AI를 특별 취급하지 않고 데이터를 가치로 바꾸는 수단 중 하나로 다룸
사람과 AI의 역할·책임·리스크를 정의·평가·관리하고, 고객·사업 가치 기준으로 분담을 수시로 다시 짬

도달 조건 (시책 하나하나는 조건일 뿐 상태 자체는 아님)

- 직원이 AI 도구를 적절히 쓰도록 정비
- 조직 데이터 자원에 적절한 접근 실현
- AI 에이전트가 업무를 수행
- AI 전제로 업무·조직 재설계

→

AI 네이티브 조직 상태

역량(Capability) · 성숙도(Maturity)로 판단

AI Enabled
AI Ready
AI First · AI Transformation

Source: 원 발표 p.55~59 (정의에는 원 발표 팀의 해석이 포함됨)

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

4.1 AI 네이티브 성숙도 5단계

KEY INSIGHT 2026년 7월 기준 다수 조직은 AI Enabled 단계에 있음 — 다음 단계는 도구 추가보다 조직·인재·데이터 준비로 결정됨

단계	이런 상태	흔한 고민
Phase 1 개인 활용	AI 활용이 개인에게 맡겨져 성과가 속인화됨	활용 여부를 판단할 기준이 없음 이용 현황 미관리(섀도 AI)
Phase 2 AI Enabled 도입·관리	도구를 도입하고 이용 현황·접근 권한을 관리 직원 AI 리터러시를 끌어올림	결과물 책임 소재가 불명확 업무는 그대로, 기존 업무 효율화에 그침
Phase 3 AI Ready 표준화	AI를 전제로 업무·의사결정 프로세스를 재검토 사람과 AI의 역할을 조직 횡단으로 정의	AI 판단 오류가 업무에 영향 부서별 AI 투자 효과를 보여 주지 못함
Phase 4 AX 평가·개선	AI 기여와 성과를 정량 파악, 효과 없으면 철수 AI 투입처를 근거로 판단	정한 역할 분담이 고정됨 효과 측정에서 멈추고 재배치로 안 이어짐
Phase 5 AI Native 최적화	변화에 맞춰 사람과 AI의 역할을 재편 생긴 여력을 가치 있는 영역에 재투자	재편 판단이 특정 인력에 의존 무거운 거버넌스와 유연한 재편의 양립

2026.07 다수 조직 위치: Phase 2

Source: Luis Gorgona, Building a Maturity Model for COBIT 2019 Based on CMMI; 원 발표 p.60~63

단계 라벨은 원 발표가 '굳이 붙이면'이라는 단서를 달아 제시한 것 · AX = AI Transformation

4.1 AX를 구성하는 세 가지 변혁 영역

KEY INSIGHT 한 영역만 바꾸면 나머지 영역이 병목이 됨 —
업무 재설계·인력 재배치·에이전트 연동 표준이 함께 움직여야 생산성이 오름

업무 변혁

작업 효율화 → 프로세스 재설계

대상 확대

사람 판단이 섞인 복합 업무도 대상에 포함

두 구조 병행

표준화·효율화와 사람·AI 조화

린 오퍼레이션

표준화·자동화로 생긴 여력을 재투자

조직 변혁

계층 축소와 인력 재배치

조직 형태

문진형 · 프로젝트형 · 통합형

AI 앰배서더

부서별 선발 인재로 병렬 추진

스킬 기반 배치

작업 요구 스킬과 직원 스킬 대조

시스템 변혁

AI 에이전트를 받는 시스템 설계

연동 표준

MCP · A2A · AP2 등 에이전트 규격

GenAIOps

업무 전문가가 개발 루프에 참여

하네스 엔지니어링

에이전트를 안전하게 돌리는 장치

Source: 원 발표 p.116~143 (Algomatic 사내 글 소개 포함); Model Context Protocol, Agent2Agent 공개 사양

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

4.1 업무와 제품 양면의 AI 활용 방침

KEY INSIGHT AI 전환은 2023년부터 이어진 시행착오 과정임 — 정해진 정답이 없어 조직마다 맞는 형태를 직접 찾아야 함

freee의 AI 활용 방침

사내외 양면에서 AI를 철저히 활용해 AI 네이티브 상태로 전환

업무의 AI 활용

사내 창조 활동·조직 활동의 동력원을
AI 우선 방식으로 재설계·재구현

개발·기획·관리 업무와 업무 환경 전반이 대상

제품의 AI 활용

고객이 쓰는 경영 플랫폼에
AI 기능을 넣어 업무 자동화 지원

회계 AI 해설·경비 신청 자동화 같은 베타 기능 제공 중

발표 자료의 접근 방식

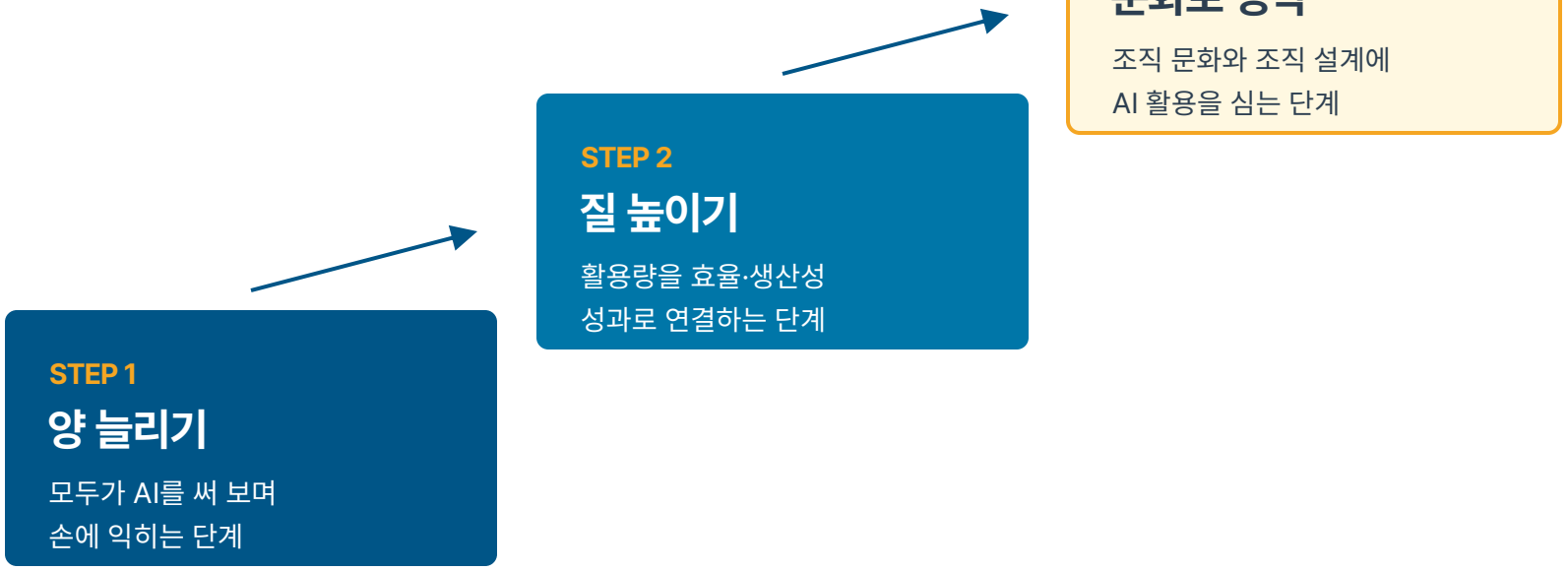
2023년부터 업무와 환경 전반을 재설계했고, 최적해가 아직 어디에도 없는 상황이라
정답 대신 사고 과정과 시행착오를 공유해 각 조직에 맞는 형태를 찾는 단서로 제시

Source: Yoshi Takeda (freee 상무집행임원 VPoE), Findy 주최 행사 발표 자료 (2025.11.13)

방침·재설계 시점은 원 발표 기준. 업무 범위 예시 줄은 보충

4.1 조직 AI 전환 3단계: 양, 질, 문화

KEY INSIGHT AI를 동료처럼 믿고 맡길 수 있는 상태가 먼저임 — 이 토대 없이 조직과 프로세스를 바꾸면 임시방편으로 끝남



토대: AI가 늘 곁에 있고 안심하고 일을 맡길 수 있는 상태

왜 양과 질로 나누나

변화 폭이 너무 큼
착수 순서를 정하기 어려움

활용의 양과 질로 나누면
단계별 목표가 분명해짐

토대가 먼저인 이유

AI 와의 신뢰 관계 없이 조직과
프로세스부터 바꾸면
임시방편으로 끝남

발표 자료는 AI를 동료와 같은
존재로 보고 신뢰를 어떻게
쌓을지를 핵심으로 봄

Source: Yoshi Takeda (freee 상무집행임원 VPoE), Findy 주최 행사 발표 자료 (2025.11.13)

3단계 구분과 토대 정의는 원 발표 기준(freee의 경우)

4.1 AI가 성과를 내는 세 영역

KEY INSIGHT AI는 인력의 성과, 반복 업무의 자동화, 제품의 강화에서 성과를 냄 — 성공한 기업은 실험과 반복으로 접근함

인력의 성과

내용

더 짧은 시간에
더 높은 품질의 결과물

성과

반복 업무의 자동화

내용

반복 작업에서 벗어나
가치를 더하는 일에 집중

자동화

제품의 강화

내용

더 관련성 높고
반응이 빠른 고객 경험

제품

AI 활용은 Software 구축이나 Cloud App 배포와 다른 새로운 방식

성공한 기업은 실험하는 자세와 반복하는 접근으로 가치에 더 빨리 도달

Source: OpenAI, AI in the Enterprise: Lessons from seven frontier companies (2025)

SECTION 4.2

사내 AI 활용 확산

체험 중심 확산과 사내 이용률

4.2 사내 AI 사용 확산 시책

KEY INSIGHT 도구 보급과 함께 안전장치·전담 조직·예산을 같이 갖춤 — 셋 중 하나가 빠지면 현장 사용이 보안 검토나 결재에서 멈춤

01 사내 AI 기반 구축

안전성 확보

입력 마스킹·출력 가드레일

이용 현황 가시화

개발자별·도구별 사용량 계측

02 AI 추진팀 신설

전문팀 구성

AI 활용 확산을 전담

엔지니어 경영진과 협업

바텀업·톱다운 양쪽에서 추진

03 CTO 직속 AI 예산

기민한 집행

번거로운 승인 절차 단축

충분한 규모

투자 성격으로 규모를 확보

양 늘리기 단계의 목표

모두가 계속 AI를 써 보며 손에 익혀, AI가 늘 곁에 있고 쓰는 것이 당연한 환경을 만드는 것

Source: Yoshi Takeda (freee 상무집행임원 VPoE), Findy 주최 행사 발표 자료 (2025.11.13)

세 시책 구성과 목표 문구는 원 발표 기준

KEY INSIGHT 비엔지니어 직군도 사내 AI 기반 위에서 앱을 만듦 — 안전한 공용 기반이 있어야 활용이 개발 조직 밖으로 넓어짐

코딩 에이전트 이용률

100% 2025년 9월 말 기준



사내 AI 기반 위에 만든 앱

1,913개 2025년 9월 말 기준

비엔지니어 직군 이용이 매우 많음

개발 조직 밖 직군도 같은 기반에서 업무 앱 제작

읽는 법 (보충)

두 수치는 활용의 양을 보여 주는 지표이고, 생산성이나 사업 성과가 오른 정도는 따로 재야 함

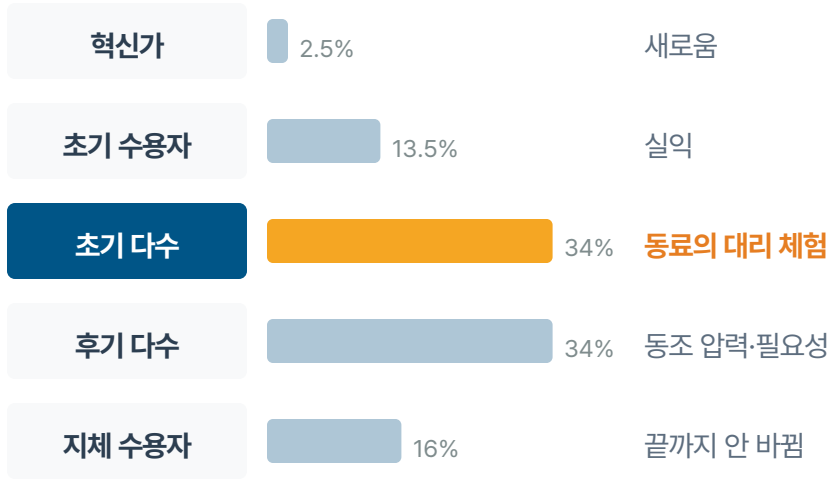
이용률 100%는 품질 향상 단계로 넘어갈 출발선이며, 측정 정의(기간·대상)는 발표 자료에 없음

Source: Yoshi Takeda (freee 상무집행임원 VPoE), Findy 주최 행사 발표 자료 (2025.11.13)

두 수치 모두 원 발표 기준, 외부 미확인. 하단 해석은 보충

KEY INSIGHT 다수 직원은 같은 업무 동료의 성공을 보고 AI를 채택함 —
눈에 보이는 성공 사례와 지원 조직이 없으면 전사 동시 도입은 정착하지 못함

Rogers 혁신 확산 — 채택 판단 기준



채택 5요소

상대적 이점 · 적합성 · 복잡성 · 시험 가능성 · 관찰 가능성

→ 전 부서 동시 도입보다 한 부서의 성공 사례를 먼저 만듦

막대는 Rogers 원전의 채택자 분포 비율

1.5선 이네이블링 팀

2선(AI 거버넌스 팀)이 감시만 하지 않고
통제 지식과 수단을 현장으로 이전해 자립을 도움
GenAIOps 에서는 업무 전문가도 개발 팀 일원
프롬프트 작성·평가 데이터 구축에 직접 참여

부서별 AI 앰배서더

업무에 정통한 인재를 부서마다 선발·육성
추진 팀 단독은 1~2개 주제가 한계,
앰배서더가 있으면 여러 주제를 병렬 추진
일반 직원에게는 최소 리터러시와 활용 틀 제공

Source: Everett Rogers, Diffusion of Innovations (1962); Citadel AI, 생성 AI 실천 가이드; 원 발표 p.94~96, p.131~132

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

4.2 생성 AI 리터러시 사내 스터디 운영

KEY INSIGHT 결과를 검토 없이 그대로 쓰는 위험은 사내 스터디로 줄임 — 위험을 알고 효율적으로 쓰는 구성원이 늘어남

위험

생성 AI 리터러시가 낮으면

결과를 해석하지 않고

그대로 쓰게 됨

대응

사내 스터디

'생성 AI 마스터 모임' 운영

리터러시 강화 프로그램

효과

위험을 이해하고

효율적으로 쓰는

구성원 사례가 늘어남

운영 포인트 **도구 도입과 리터러시 교육을 함께 진행**

발표 자료는 이 스터디를 생성 AI 활용 위험에 대한 대응책으로 소개함

Source: 원 발표 자료(원커리어, 2025.04.23) 23쪽 요지

4.2 도구가 있어도 현장 활용이 멈춤

KEY INSIGHT 현업 직원이 자기 업무에서 쓸 곳을 찾지 못함 — 도구 보급만으로는 활용이 늘지 않음

경영진

전사 AI 확산 지시
활용 확대 목표 제시

→

AI 담당

일단 써 보라는 권유
도구 사용 자체를 강조

→

현업 관리자

적용할 업무 판단 곤란
어디에 쓸지 되묻는 상태

→

현업 직원

업무 적용 감각 부족
쓸 곳을 찾지 못한 상태

Blank Canvas 문제

무엇이든 할 수 있지만 어디서 시작할지 모르는 상태

AI 도구는 손에 있는데 현장에서 쓰이지 않는 상황을 어떻게 움직일지가 과제

Source: 일본 오라클 OCI Platform COE 본부 솔루션즈 아키텍트, AI 주도 AI 확산 활동(AAAP) 발표 자료, 2026 2쪽

4.2 교육 중심 방식과 체험 중심 방식 비교

KEY INSIGHT AI 와의 대화 한 번으로 업무 정리·사례 발굴·장벽 확인이 함께 이뤄짐 — 조작법 강의만으로는 현장 적용이 따라오지 않음

기존 방식

교육 중심

교육에서 조작 방법을 설명

프롬프트 작성법을 강의

현장 업무에 적용하지 못함

AI 주도 방식

체험 중심

AI와 대화하는 체험을 설계해 쓰면서 장벽을 줄임

대화 한 번으로 함께 얻는 것

- 업무를 말로 정리
- 활용 사례 발굴
- 장벽 확인

Human-in-the-loop 원칙

AI는 대화 진행·정리·통합을 맡고, 의사결정과 최종 판단은 사람이 맡음

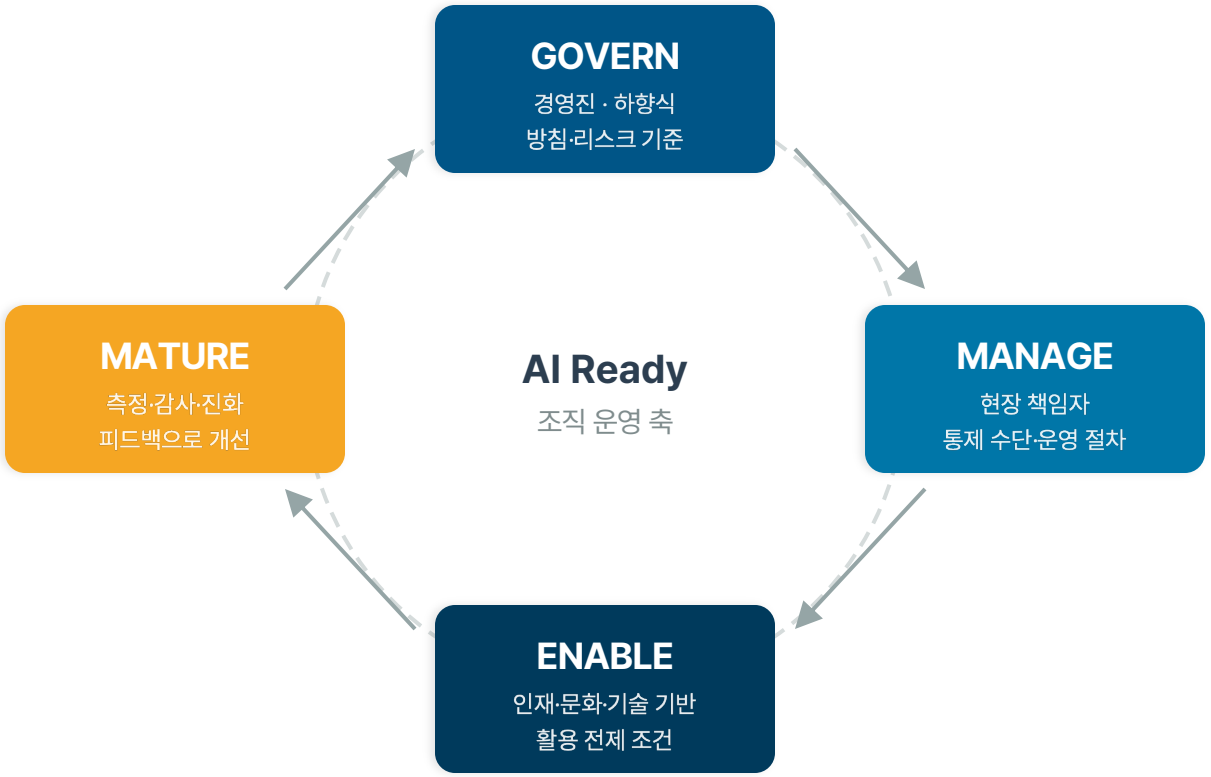
Source: 일본 오라클 OCI Platform COE 본부 솔루션즈 아키텍트, AI 주도 AI 확산 활동(AAAP) 발표 자료, 2026 5쪽

SECTION 4.3

AI 거버넌스와 책임

AI Ready 네 가지 기준과 책임 공백

KEY INSIGHT 네 축은 서로 이어진 순환 구조임 —
각 축이 Agility·Reliability·Integrity 성과로 이어질 때 지속적 경쟁 우위가 생김



축별 성과 연결

GOVERN — Agility
비즈니스 가치 창출
빨리 정하고 많이 시도하고 빨리 적응

MANAGE — Reliability
리스크 최적화
흔들리지 않고 멈추지 않고 계속 운영

ENABLE — Reliability
자원 최적화
속인 의존을 줄이고 공통 기반을 제공

MATURE — Integrity
지속적 경쟁 우위
정책과 운영 실태를 일치시켜 신뢰 축적

Source: 원 발표 p.84, p.104~106; NIST, AI Risk Management Framework (GOVERN·MAP·MEASURE·MANAGE)

4.3 GOVERN: 실패 원인은 부족보다 괴리

KEY INSIGHT 규정과 위원회가 있어도 경영진 행동이 다르면 통제가 작동하지 않음 — 경영진이 AI 활용 실적과 실패 사례를 먼저 공개해야 함

말과 행동의 괴리

- 명시한 방침보다 실제 행동이 더 강하게 따라 하게 됨
- 경영진의 언행 불일치는 신뢰를 크게 훼손

방침과 실천의 괴리

- 규정·체계·위원회는 있으나 운영되지 않음
- 형식은 대외용, 실무는 따로 도는 디커플링

수단과 목적의 괴리

- 교육·평가를 해도 리스크 감소 효과를 검증 안 함
- 지표가 목표가 되는 굿하트 법칙이 조직에 확대

변화와 의미 체계의 괴리

- 제도만 갖추고 전환 과정을 설계하지 않으면 실패
- 형식적 통제 충족이 추진 조직을 지치게 함

통제 구조 참고 — IIA 3선 모델

1선 업무 부서 — 제품·서비스 제공과 리스크 관리

2선 리스크 관리 부서 — 전문 지식·지원·감시

3선 내부감사 — 독립적인 보증과 조언

Source: The IIA, Three Lines Model (2020); 일본 경제산업성 시스템 관리 기준; 원 발표 p.86~87

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

KEY INSIGHT 권한·승인·사전 확인·정책·모니터링 경계를 사람이 설계해야 작업을 넘길 수 있음 — 결과는 정상 동작 근거와 함께 받음

1

최소 권한

AI·자동화에 주는
권한은 필요한
범위만 부여

2

승인 단계

운영 환경 변경은
반드시 사람 승인을
거침

3

변경 전 확인

what-if로 적용 전에
무엇이 바뀌는지
미리 확인

4

Azure Policy

넘으면 안 되는 선을
가드레일로 강제

5

모니터링

변경 후 동작과
비용을 계속 관찰

AI 작업 결과와 함께 받을 근거

동작한다는 주장 대신 올바르다는 증거를 함께 받음

- ✓ what-if 결과(무엇이 바뀌는지)
- ✓ 테스트·검증 기록
- ✓ 정책 준수 확인 결과
- ✓ 배포 후 모니터링 현황

what-if는 기존 리소스를 바꾸지 않고 변경 내역만 미리 보여 줌 — 리소스 그룹·구독·관리 그룹·테넌트 범위 지원

Source: 원 발표 p.13 · Microsoft Learn "Bicep deployment what-if operation" · 원 발표가 참고한 Simon Willison's Weblog

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

4.3 AI 위임을 막는 책임 공백과 맥락 부족

KEY INSIGHT 잘못된 결과의 책임은 결국 이용자·사업자가 짐 —
결제·발송·삭제처럼 되돌릴 수 없는 행동은 사람 승인 단계를 두어야 함

① 책임 공백

AI는 결과에 책임지지 않음

- 블랙박스성 — 내부 판단 과정이 보이지 않음
- 비결정적 동작 — 같은 입력에도 결과가 달라짐
- 사람의 인지 범위를 넘는 행동
- 잘못되면 책임은 이용자·사업자에게 전가

되돌릴 수 없는 행동

결제 · 발송 · 공개 · 삭제 · 변경

② 맥락 부족

조직이 AI를 받아들일 준비가 덜 됨

- 사람의 지시는 늘 모호함 — '알아서 해 줘'
- 데이터를 포함한 정보 자원이 정비되지 않음
- 맥락이 많은 조직일수록 위임 가능한 범위가 좁음
- 복잡해질수록 통제 지점이 급격히 증가

정비가 필요한 대상

지시의 명확성 · 정보 자원 · 업무 기록

Source: 원 발표 p.19~21

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

4.3 한국 조직 적용 시 점검 항목

KEY INSIGHT AI 기본법이 2026년 1월 시행돼 계도기간이 진행 중임 — 과태료 부과 전에 고영향 AI 해당 여부와 책임 구조를 정해야 함

보충 한국 제도 현황

2026.01.22

인공지능 기본법 시행

1년 이상

계도기간 운영 예정 — 과태료는 그 뒤

사업자 구분

개발사업자 · 이용사업자

이용사업자

타사 AI API를 연동해 서비스하는 기업도 해당

고영향·생성형

투명성·안전성 확보와 고지 의무

세부 기준은 법제처 원문·과기정통부 가이드라인으로 확인

성숙도 단계 진단

현재 Phase를 5단계 표로 판정 — 대부분 Phase 2 예상

섀도 AI 파악

이용 현황·접근 권한을 목록화하고 승인 절차 마련

사람 승인 지점

결제·발송·삭제 등 되돌릴 수 없는 행동부터 정의

앰배서더·지원 팀

부서별 앰배서더와 1.5선 이네이블링 팀 지정

지표 측정 시작

Four Keys에 시도·위임·처리량·재작업 지표 추가

Source: 법무법인 세종(shinkim.com) 뉴스레터, AI 기본법 해설; 원 발표 p.60 성숙도 표

이 장 전체가 보충 — 원 발표에는 없는 한국 제도 정보

4.3 AI 도입 뒤 생기는 세 가지 적응 과제

KEY INSIGHT 검증·운영·책임 부담이 풀리지 않으면 편익보다 비용이 커짐 — 고성과 조직일수록 업무 흐름 재설계에 먼저 투자함

검증 비용 증가

- AI 출력 확인에 전문성과 시간이 듦
- 사용량이 늘수록 확인 부담도 비례해 증가
- 인수인계 때 정보가 늘 부족함

새 운영 업무 발생

- 프롬프트·모델 갱신을 따라가기 어려움
- 비용·사용량을 계속 감시해야 함
- 보안 대응 요구가 추가됨

책임과 신뢰의 재정의

- 책임 분계점이 명확히 정의되지 않음
- AI 성능의 과대·과소평가 대응 부족
- 기존 의사결정 흐름과 맞지 않음

근거: Lisanne Bainbridge, Ironies of Automation(1983) — 자동화할수록 사람에게 남는 감시·예외 처리가 더 어려워짐

보충 **McKinsey, The state of AI in 2025 (2025.11)**

약 88%
AI를 한 개 이상 업무에 정기 활용하는 조직

약 6%
EBIT 5% 이상을 AI 기여로 보는 고성과 조직

약 3배
고성과 조직의 업무 흐름 근본 재설계 비율

Source: 원 발표 p.67~71; McKinsey, The state of AI in 2025: Agents, innovation, and transformation (2025.11)

원 발표: Algomatic 미야와키 슌페이, AI 네이티브 조직을 다시 묻다 (2026.07)

SECTION 4.4

팀 규모와 인력 판단

6명 팀을 2명으로 줄일지에 대한 판단

KEY INSIGHT 산출물이 그대로라는 가정이 이 구상의 전제 — 비율이 1/2 이든 1/100 이든 같은 구조로 따져 볼 수 있음

Before

6명 팀

산출물 1

AI 없이 6명이 함께 만드는 기준 상태

AI 최대 활용

After (경영진이 기대하는 그림)

2명과 AI

AI

산출물 1 유지 가정

남은 4명 재배치

프로젝트 A

프로젝트 B

같은 산출물을 1/3 인건비로 내고 남은 인력은 다른 매출에 투입

발표 자료의 답: 6명 모두가 매우 유능하면 단기적으로는 아마 가능함(이론상)

남는 질문은 그 2명 팀이 계속 유지되는가 — 휴가·퇴사·다음 담당자 육성까지 견디는가

비율은 자기 회사 사정에 맞게 바꿔 읽어도 같은 구조가 됨

Source: 원 발표 자료 20~22쪽

원 발표는 AI를 최대한 잘 활용해 인원을 1/3로 줄일 수 있다고 가정하고, 현실성은 따지지 않은 채 구조만 살핌

2명 팀은 시니어 의존을 키움

KEY INSIGHT 단기 속도는 나오지만 팀으로서 버티는 힘이 약함 — 다음에 같은 판단을 할 사람이 늘지 않음

1 휴가·퇴사에 취약

1명이 장기 휴가·육아휴직·병가·
퇴사하는 순간 업무가 한꺼번에 멈춤

인원 수 이상의 위험

2 시니어가 빠질 수 없는 구조

판단과 맥락이 AI를 잘 쓰는
사람에게 쏠림

그 시니어가 없으면
팀이 돌아가지 않음

3 다음 담당자가 자라지 않음

주니어·미들이 설계 판단·리뷰·
장애 대응·고객 응대를
옆에서 겪을 기회가 줄어듦

같은 판단을 할 사람이 늘지 않음

효율화했다고 생각했지만 실제로는 시니어 의존을 압축했을 뿐

· 짧게 보면 빠르지만 팀의 내구력이 낮음 — 시니어 본인도 지치고 조직도 확장되지 않음

· 산출물은 만들어도 강한 팀을 다시 만들어 내지는 못함

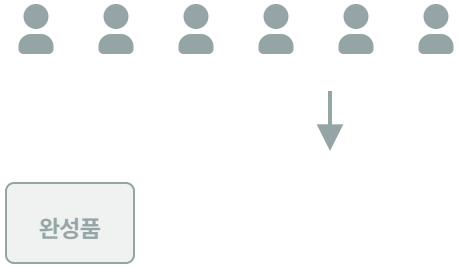
Source: 원 발표 자료 24~25쪽

세 가지 위험과 설명은 원 발표 도식의 요지를 정리함

KEY INSIGHT 2명과 AI로 같은 성과를 내기보다 6명과 AI로 성과를 키우는 편이 오래감 — 줄인 인원만큼 위험도 커짐

기준 상태

6명, AI 없음



완성품 1

사람이 하는 만큼 결과가 나옴

업계가 향하는 방향

2명과 AI

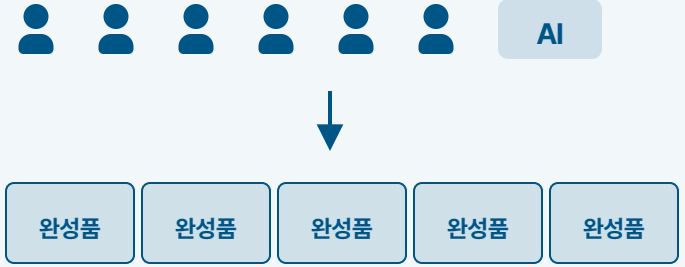


완성품 1

인원을 줄이고 같은 성과를 유지

발표 자료가 다루는 방향

6명과 AI



완성품 여러 개

인원이 늘면 성과가 배로 늘어남

목표: 사람을 늘렸을 때 성과도 함께 늘어나는 구조를 설계함

발표 자료: KAG는 이 구조를 여러 팀에서 운영 중이며, 그 핵심이 학습하는 팀과 피처 몹

Source: 원 발표 자료 27~29쪽

KAG가 여러 팀에서 이 구조를 운영 중이라는 내용은 원 발표 기준, 외부 미확인

KEY INSIGHT 비개발자도 실용 앱을 만드는 사례가 나오면서 팀을 줄여도 된다는 판단이 힘을 얻음

개발사의 검토

AI 이용료에 투자한 만큼 인건비를 줄일 수 있는지 검토

- 1인 다역
- 6명 팀을 2명으로 축소
- 여러 프로젝트 겸임
- 남는 시간에 다른 업무

Anthropic 해커톤 1위

Claude Code 해커톤 (Built with Opus 4.6)

- 1위: 캘리포니아 변호사
- 주택 인허가 검토 앱 CrossBeam
- 수상자 5명 중 4명이 비개발자
- 순환기내과 의사·도로 기술자도 수상

SNS에 늘어난 사례

1명이 AI 에이전트 6개를 운영하며 팀처럼 일한다는 게시물

- 아침에 구현 방침 5가지 생성
- 점심까지 전부 프로토타입
- 오후에 1개만 남기고 폐기
- 회의 전에 4개를 먼저 버림

발표 자료가 던진 질문: 이제 소프트웨어 엔지니어도, 팀도 필요 없는가

Source: 원 발표 자료 6~12쪽; claude.com/blog/meet-the-winners-of-our-built-with-opus-4-6-claude-code-hackathon

해커톤 내용은 Anthropic 발표로 확인(보충). SNS 사례는 원 발표가 2026년 3~5월 게시물을 재구성한 것으로 개별 출처 미확인

KEY INSIGHT 업무 문제를 아는 사람이 AI로 해결 도구를 직접 만듦 — 무엇을 만들지 정하는 능력이 결과를 가름

Built with Opus 4.6 해커톤 입상자 5명의 본업

1위				
변호사	**순환기내과 의사**	**도로·인프라 담당**	**전자 음악가**	**소프트웨어 엔지니어**
비엔지니어	비엔지니어	비엔지니어	비엔지니어	엔지니어

1위 Mike Brown(변호사) — CrossBeam

캘리포니아 건축 허가가 반려됐을 때 제출하는 회신 문서를 AI로 작성하는 도구

수상 소감 요지: 코드를 한 줄도 쓰지 않고 우승해 본인도 놀라움

업무 절차를 잘 아는 사람이 문제를 정의하고, 구현은 AI가 맡은 사례

일본 대상 행사 Built with Claude: Japan 2026도 2026.11.2~10에 열릴 예정(발표 자료 안내)

Source: Anthropic 'Meet the winners of Built with Opus 4.6' · 'Built with Claude: Japan 2026' — 원 발표 인용

입상자 직업·사례는 원 발표 기준, 미확인. 원 발표의 입상자 사진은 싣지 않음

KEY INSIGHT 대부분 AI 이전부터 있던 문제임 — AI가 작업을 빠르게 하면서 그 문제가 병목으로 드러남

리뷰 부담

늘어난 변경을 사람이 읽고
승인하는 데 시간이 몰림

인지 부하

한 사람이 파악해야 할 코드와
맥락의 양이 커짐

기술 부채

충분히 이해하지 못한 코드가
빠르게 쌓임

특정인 의존

담당자만 아는 영역이 늘어
휴가·이동 때 업무가 멈춤

구성원 육성

AI가 일을 대신하면서 신입이
배울 기회가 줄어듦

문서화

결정 배경과 맥락이
기록으로 남지 않음

DORA 2025 — AI는 증폭기

잘하는 조직의 **강점**과 어려움을 겪는 조직의 **문제**를 함께 키움

발표 자료자 해석: 이 문제들은 AI 이전부터 있었고, AI로 작업이 빨라지면서 병목이 되기 쉬워짐

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표 · Google DORA, State of AI-assisted Software Development 2025. 각 문제의 풀이 문구는 보충

SECTION 4.5

AI 시대 리더의 역할

병목이 되기 쉬운 관리자의 역할

KEY INSIGHT 경영진과 중간 관리자가 변화를 늦추는 지점이 되기 쉬움 — 코칭과 장애 제거로 받치면 조직 전체가 빨리 바뀜

먼저 자각할 것

경영진과 중간 관리자가 변화의 병목이 되기 쉬움

반대로 관리자가 변화를 돕는 쪽에 서면 팀과 조직의 변화가 빨라짐

발표 자료에서는 덧붙이는 이야기로 다룸

관리자에게 필요한 역량 — 팀을 지원하는 힘

팀 코칭
팀이 스스로 일하는 방식을 정하도록 질문하고 도움

변화를 막는 요인 제거
규정·결재·평가 기준처럼 팀이 못 바꾸는 장애를 치움

변화 촉진
새 방식을 시도할 시간과 범위를 열어 줌

팀 생산성을 의심하지 않고 받쳐 줌
몹으로 사람 수 대비 작업량이 줄어 보여도 아웃컴으로 판단

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07). 각 역량의 풀이 문구는 보충

4.5 관리자 관점: X이론과 Y이론

KEY INSIGHT 제품 개발은 목적과 재량을 맡기는 Y이론과 맞음 — X이론식 통제는 안전·법령·보안·장애 대응 영역에 남김

X이론

사람은 내버려 두면 게을러지기 쉬움

- **세밀하게 관리** — 체크리스트로 확인
- **지시·명령** — 무엇을 할지 위에서 정함
- **승인이 필요** — 작은 판단도 결재를 기다림
- **규칙 엄수** — 정해진 방식만 허용

팀의 결과

지시를 기다리는 팀 — 아이디어가 있어도 말하지 못함

Y이론

사람은 스스로 생각하고 책임지며 성장하고 싶어 함

- **목적과 재량을 맡김** — 사용자에게 줄 가치를 공유
- **질문을 던짐** — 답 대신 생각할 거리를 줌
- **의사결정을 위임** — 기술 판단은 팀이 내림
- **실험을 응원** — 가설 → 검증 → 학습을 지원

팀의 결과

스스로 배우는 팀 — 시도하고 지식을 공유함

Source: 원 발표 자료 43~44쪽; 더글러스 맥그리거, The Human Side of Enterprise(1960)

원 발표는 강한 팀의 전제 조건으로 팀 바깥 관리자의 관점 변화를 꼽음

4.5 현장 소통: 질문과 책임 위임으로 사고 유도

KEY INSIGHT 답을 바로 주면 생각할 기회가 사라짐 — 질문과 위임이 스스로 생각할 기회를 늘림

생각을 이끄는 본질 질문

"애초에 왜 이렇게 설계했지?"

상사가 답을 주는 대신 일부러 질문을 던짐

구성원이 한 번 멈춰 깊이 생각할 계기 마련

겨냥하는 힘: 비판적 성찰

자율을 이끄는 책임 위임

업무 책임을 구성원에게 맡김

맡은 일에 대한 주인 의식이 높아짐

스스로 생각하고 움직일 기회를 의도적으로 늘림

겨냥하는 힘: 자율적 사고

제도 밖 일상 대화에서도 같은 훈련을 이어 감 — 답을 주는 관리자에서 질문하는 관리자로 역할 전환

Source: ONE CAREER Inc., Forkwell 행사 발표(2026-04-24)

4.5 AI로 생긴 여유 시간의 투자처

KEY INSIGHT 여유 시간의 쓰임을 팀이 함께 정하고 실행함 — 기능 추가에만 쓰면 아웃컴은 그대로임

AI 이후 몹의 하루

프롬프트 작성 · 결과 리뷰

AI로 생긴 여유 시간 — 어디에 쓸 것인가

피할 것

- 기능을 계속 추가하는 데 다 씀
- 개발 속도를 높이는 일에만 집착

아웃풋(작업량)은 늘어도 아웃컴은 그대로일 수 있음

할 것

- 아웃컴을 높이려면 무엇이 필요한지 생각
- 그곳에 시간을 투자

예(보충): 사용자 관찰, 기술 부채 정리, 구성원 학습

여유 시간을 어디에 쓸지 **팀이 함께 생각하고 판단하고 실행**하는 일 자체가 여유 시간의 쓰임

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표(2026-03-07). 할 것의 예시는 보충

4.5 AI는 팀의 강점과 약점을 함께 키움

KEY INSIGHT 약한 팀은 AI로 더 빨리 무너지고 강한 팀은 더 빨리 배움 — 성과 차이는 도구보다 팀 상태에서 나옴

약한 팀 — 빠르게 무너짐

- 모호한 사양
- 형식적인 리뷰
- 약한 테스트
- 팀 안에 갇힌 지식

AI는 팀을 증폭함

강한 팀 — 빠르게 배움

- 작게 시도
- 리뷰로 학습
- 테스트로 보호
- 실패 공유

AI 성과를 높이는 조건은 강한 팀

· 발표 자료: 같은 AI를 써도 약한 팀은 문제를 더 빨리 키우고, 강한 팀은 학습 속도를 높임

· 보충(DORA 2025): 큰 성과는 도구 자체보다 내부 플랫폼 품질·업무 흐름의 명확성·팀 정렬에서 나옴

Source: 원 발표 자료 33쪽; dora.dev/research/2025; research.google/pubs/dora-2025-state-of-ai-assisted-software-development-report

보충: DORA 2025는 전 세계 기술 실무자 약 5,000명 설문과 100시간 이상의 인터뷰를 바탕으로 AI를 조직의 강점·약점을 키우는 증폭기로 정리함

4.5 에이전트를 맡기고 검수하는 Agent boss

KEY INSIGHT 리더의 82%가 12~18개월 안에 디지털 노동력 도입을 계획함 — 구성원의 역할은 직접 실행에서 에이전트에게 맡기고 검수하는 쪽으로 이동

82%

12~18개월 안에 디지털 노동력 도입 계획

리더 응답

78%

AI 관련 신규 직무 채용 검토

리더 응답

1 만들기

자기 업무를 맡을 에이전트를 직접 만들거나 고름

2 맡기기

목적과 완료 기준을 정해 과업을 위임함

3 관리·검수

결과를 판정하고, 사람 대 에이전트 비율을 결과 지표와 함께 봄

Source: Microsoft, 2025 Work Trend Index: The Frontier Firm is born(2025.04)

SECTION 4.6

AX 인재상과 채용 기준

AX 인재 공통 역량과 채용 평가표

4.6 AX 인재 공통 역량 다섯 가지

KEY INSIGHT 다섯 역량은 직무와 관계없이 적용됨 — 직무별 차이는 같은 역량이 어떤 업무에 쓰이는지에서 나타남

역량	행동 단위 정의	직무별로 쓰이는 예
목적 정의	무엇을 왜 맡길지, 어떤 결과를 성공으로 볼지 먼저 정함	마케터는 캠페인 목적, 개발자는 완료 조건
도메인 깊이	업무 대상과 판정 규칙을 알고 결과의 옳고 그름을 가려냄	경리는 회계 판단, 영업은 고객 결정 근거
위임과 검증	AI가 잘하는 과업과 못하는 과업을 나눠 맡기고 결과를 확인함	마케터는 초안, 개발자는 코드 변경을 검증
맥락 설계	에이전트가 참조할 자료·규칙·예시를 골라 업무 맥락으로 구성함	금칙어 목록, 코드 규칙 파일
결과 책임	산출물 개수보다 손익·고객 결과로 성과를 설명함	발행 편수 대신 리드 전환, 코드량 대신 결함 감소

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.1절(2026.10)

4.6 채용 평가 항목의 가중치 조정안

KEY INSIGHT 채용 실무자도 비판적 사고(73%)를 1위, AI 활용 역량을 5위로 둠 — AI 도구 숙련은 기본 자격으로 두고 문제 정의·결과 검증 가중치를 높임

평가 항목	기존 관행	조정안	면접에서 확인하는 방법
AI 도구 사용 경험	높음	중간(기본 자격)	사용한 도구 목록보다 맡긴 과업과 결과를 설명하게 함
문제 정의	중간	높음	모호한 과제를 주고 목표와 성공 기준을 적게 함
결과 검증	낮음	높음	오류가 섞인 AI 산출물을 주고 결함을 찾게 함
도메인 지식	중간	높음	해당 업무의 판정 규칙을 설명하게 함
결과 책임	낮음	중간 이상	과거 성과를 손익·고객 결과로 설명하게 함

Source: Korn Ferry, TA Trends 2026(2025.10) · MSAP.ai 백서 6.1.2절 — 기존 관행은 예시

4.6 엔지니어 채용 기준이 사업 기여 중심으로 바뀜

KEY INSIGHT 채용 담당자 약 40%가 요구 스킬이 바뀌었다고 답함 — 코딩 능력만으로는 AI와 차별화하기 어려움

엔지니어 채용 버블기 (2015~2022)		경기 둔화 + AI 확산 (2026)
정규직 엔지니어 머릿수 확보가 최우선 지금의 GPU 확보와 같은 위치	→	**AI를 활용하는 즉시 전력**
스킬 중심 · 직무형 채용	→	**사업 기여 · 직무 경계를 넘는 역량**
잠재력 중심 채용	→	**소통 능력 중시**
워라밸 중시도 수용	→	**몰입해서 일하는 태도**

</>

채용 담당자 응답

약 40%

"요구하는 스킬이 바뀌었다"

코딩 스킬 하나만으로는
AI와 비교되어 가치가 낮아짐

조사 주체·표본은 발표 자료에
명시되지 않음

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07)

4.6 2026년 채용 기준은 소통과 사업 기여

KEY INSIGHT 기술 역량만으로는 채용 문턱을 넘기 어려워짐 — 직무형 고용 10년 뒤 소속 중심 고용과 인력 감축이 함께 진행됨

2025년 이전

M/M(맨먼스) 발상

- 머릿수를 채우면 생산성이 오른다고 봄
- 직무형 고용 권장 — 2020년 전후 과열
- JD를 정의하고 급여와 연결
- 목표: 인력 유동화

2026년 가을 현재

함께 일하기 좋은 사람 우선

- **소통 능력 최우선**
- **영업직 우선 채용**
- **사업 기여 우선**
- **직무 경계를 넘나드는 성향**

발표 자료 표현: "좋은 녀석" — 성격보다 협업 역량을 가리킴

흐름 직무형 고용 10년 → 소속 중심 고용(멤버십형)으로 회귀 + 인력 감축이 동시에 진행

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07)

M/M: 투입 인원 × 기간으로 공수를 계산하는 방식

4.6 대인 커뮤니케이션이 두 번째 평가 기준이 됨

KEY INSIGHT 개발을 AI에 맡길수록 사람끼리의 조정·합의가 남음 — 하드 스킬 차이가 줄어 소프트 스킬로 차이가 남

1차 기준 하드 스킬

AI 활용 능력을 포함한 기술 역량

여전히 첫 번째 관문

AI 보조로 사람 간 차이가 줄어듦

→

2차 기준 소프트 스킬

커뮤니케이션 능력 등

개발을 AI에 맡기면 사람에게는
사람끼리의 대화가 남음

차이가 드러나는 자리

사람에게 남는 대인 업무

비즈니스와 기술 연결

요구를 기술 과제로 옮겨 설명

조정

일정·범위를 관계자와 맞춤

합의

결정 사항을 함께 확정

1년 차에게 여기까지 요구하지는 않지만, 첫걸음으로 지적을 그대로 받아들이는 사람이 선택받음

결과에 책임지기로 했다면 소통 방식도 거기에 맞춰 바꿈

Source: 원 발표(스리세이크, 2026.02.04) 대인 커뮤니케이션 쪽

대인 업무 세 카드의 보조 설명은 보충

4.6 컨설팅과 사업회사의 인력 구성 변화

KEY INSIGHT 두 업종 모두 즉시 전력과 사내 육성에 집중함 — 중간 경력층의 외부 채용 수요가 줄어듦

컨설팅 — 조사·정리 업무의 가치 하락

2025년 이전

조사·정리가 핵심 가치

BPO 사업 강화와 인력 이동

2026년 가을 현재

조사·정리를 AI가 맡기 시작, BPO 에도 AI 진출

일부 대형사는 신입 채용을 줄이고 즉시 전력에 집중

전망

AI 도입에 따라 적정 인원 자체가 달라짐

고객의 요구를 받아 주는 대응 인력 수요는
남아 있으나 규모 유지는 불확실

사업회사 — AI 활용 성공 기업의 움직임

인력 적정화

AI로 대체 가능한 업무부터 인원 조정

조기퇴직 모집 병행

소수 신입 채용으로 연령 구성 조정

소통 능력은 기본 조건

사업에 관심 있는 인재를 제품 엔지니어 ·
BizDev · GTM 담당으로 육성

CS 석·박사를 장래 서비스 책임자로 육성

중간 경력층은 외부 채용보다 사내 육성

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07)

BPO: Business Process Outsourcing(업무 위탁) · GTM: Go-To-Market(시장 진입 전략) · CS: Computer Science

SECTION 4.7

주니어 육성 공백

시니어는 빨라지고 주니어 채용은 줄어듦

4.7 시니어는 빨라지고 주니어는 느려짐

KEY INSIGHT 판단력이 있어야 AI가 속도로 이어짐 — AI 노출 직군의 22~25세 고용은 이미 기대 경로보다 크게 낮음

AI boost — 시니어

경험에 기반한 판단 기준으로 AI 결과를
빠르게 검증하고 고침

AI가 속도를 크게 끌어올림

AI drag — 주니어

판단 기준 없이 AI 결과를 받아들여
오류를 놓치고 재작업이 늘어남

AI 사용이 오히려 역효과

고용 데이터

−7.7%

AI를 도입한 기업의
주니어 고용 감소폭

SSRN 5425555 · 발표 자료 인용

−13%

AI 노출 직군 22~25세 고용
상대적 감소 · 경력자는 안정·증가

Stanford Digital Economy Lab · 2025-08 · 발표 자료 인용

−19% 최신 보충

같은 연구 개정판: 기대 경로 대비
22~25세 고용 격차 확대

Stanford Digital Economy Lab · 2026-08 개정

Source: 원 발표 p.23–24 · Stanford Digital Economy Lab, Canaries in the Coal Mine?(2025-08 초판, 2026-08 개정 · 초기 지표)

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

4.7 주니어 채용을 줄이면 시니어가 부족해짐

KEY INSIGHT 몇 년 뒤 경쟁사 인력을 높은 연봉으로 영입하는 비용이 생김 — 육성 경로를 지금 설계해야 피할 수 있음

인재 파이프라인(Talent Pipeline)이 끊기는 순서

1단계	2단계	3단계	4단계
주니어 생산성 저하	**주니어 채용 축소**	**차세대 시니어 미육성**	**외부 시니어 고액 영입**
판단 기준이 없는 주니어는 AI를 써도 속도가 나지 않음	당장 비용이 덜 드는 쪽을 택해 신입 채용을 줄임	경험을 쌓을 주니어가 없어 몇 년 뒤 시니어 후보가 비어 있음	경쟁사에서 시니어를 높은 연봉으로 데려와야 하는 상황

근거 논문: Redefining the Software Engineering Profession for AI

Mark Russinovich(Azure CTO) · Scott Hanselman 공저, Communications of the ACM

논문 경고 요지: "인재 파이프라인은 무너질 것" — 대안으로 정식 육성 직무(프리셉터) 제안

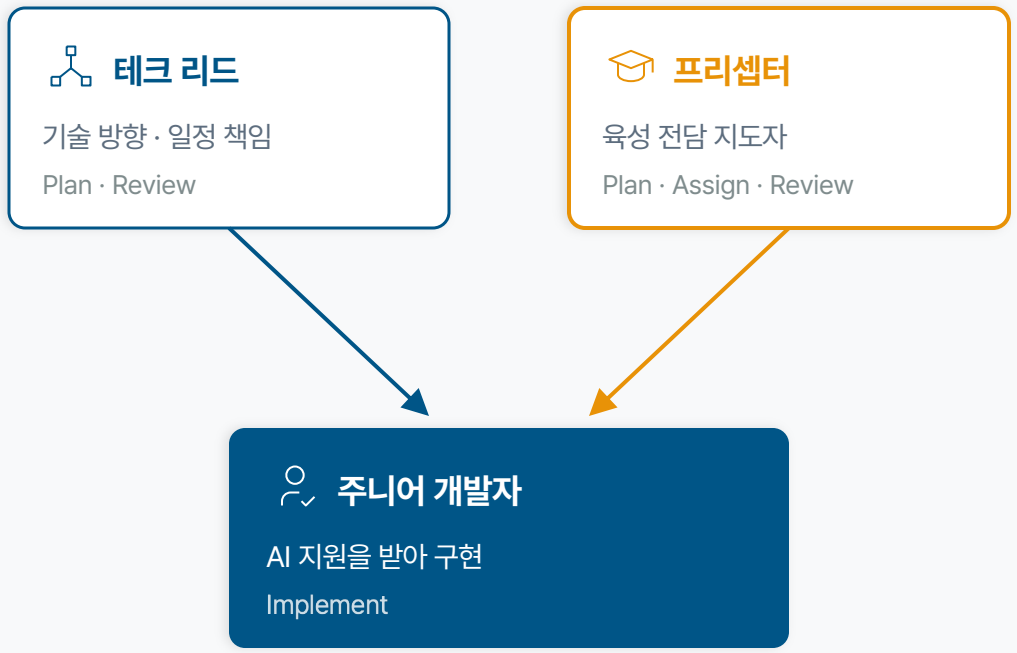
Source: 원 발표 p.25 · Communications of the ACM, DOI 10.1145/3779312

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

4.7 프리셉터 제도: 육성 전담 지도자

KEY INSIGHT 코드 산출보다 엔지니어 육성 성과로 평가받는 정식 직무가 필요함 — 별도 역할로 둬야 육성이 일정 압박에 밀리지 않음

AI 시대의 지도 체계



프리셉터가 과제를 나누어 맡기고 결과를 함께 리뷰

유래와 위상

간호 분야의 프리셉터(지도 간호사) 제도에서 차용
모두가 맡지 않는, 정식으로 정의된 직무

평가 기준

신입을 제 몫을 하는 엔지니어로 키웠는가
연간 평가: 지도 역할 유지 여부·지도자 교육 필요 여부

조직 배치

테크 리드 옆에 육성 전담 프리셉터를 별도로 둠
기술 일정 책임과 육성 책임을 한 사람에게 함께 맡기지 않음

인용 요지 — Scott Hanselman

프리셉터가 내놓는 결과물은
제 몫을 하는 엔지니어 — 코드 산출량으로 평가하지 않음

Source: 원 발표 p.26–28 · InfoQ Architects Podcast(Scott Hanselman) · Communications of the ACM 논문

4.7 한국 IT 조직의 신입 육성 개선안

KEY INSIGHT 우리가 먼저 신입 과제를 기획부터 고객 전달까지 한 주기로 설계해야 함 — 코드 작성만 맡기면 AI가 대신하는 일만 배움

보충 — 발표 자료 사례를 한국 IT 조직에 적용할 때의 검토안

1 신입 과제를 분석·구현·출시·고객 안내까지 한 주기로 설계

근거: STORES 신입 연수는 데이터 분석부터 점주 안내 메일까지 한 사람이 이어서 맡음
코드 작성만 맡기면 AI가 대신하는 일만 익히게 됨

2 Lint·테스트·AI 코드 리뷰 같은 검증 체계를 온보딩에 포함

근거: 개발자 46%가 AI 결과 정확성을 불신 (Stack Overflow 2025)
신입이 첫 주부터 검증 도구를 직접 다뤄 봐야 AI 변경을 판단할 수 있음

3 성과 지표를 코드량에서 배포 빈도·변경 실패율 같은 전달 지표로 전환

근거: DORA 지표는 배포 빈도와 변경 실패율을 함께 봄
AI로 코드량은 쉽게 늘어나므로 코드량 지표는 성과를 구분하지 못함

Source: (보충) 원 발표 사례 기반 한국어판 검토안 / Stack Overflow Developer Survey 2025 — survey.stackoverflow.co/2025/ai
(보충) DORA 2025 — infoq.com/news/2025/09/dora-state-of-ai-in-dev-2025

4.7 AI 대화와 사람 협업의 마찰 차이

KEY INSIGHT AI 대화만으로는 의견 충돌과 공동 책임을 겪지 않음 — 신입이 협업 역량을 기를 기회가 그만큼 줄어듦

	AI와 대화 혼자 빠르게 답을 얻는 환경	사람과 협업 관점이 다른 동료와 함께 일하는 환경
마찰 비용	마찰 비용이 낮음	**의견이 부딪침**
관계	관계가 깨질 걱정이 없음	**프로젝트 뒤에도 관계가 이어짐**
책임	결과를 떠안지 않음	**결과를 함께 떠안음**

Source: PBL Pub 2026 Keynote(왜 키우기 어려워졌는가). 열 머리의 환경 설명은 보충

4.7 인재 육성 전제가 바뀐 네 가지 이유

KEY INSIGHT 역할이 유동적일수록 기존 OJT 방식이 통하지 않음 — 개인의 강점·약점을 다시 언어화해 기준을 맞춰야 함

01

구성원 특성의 다양성

역할·스킬·경험이 원래
제각각이고 상사와 같은 특성을
가진 경우가 드묾

02

육성 정답의 소멸

누구도 겪어 보지 못한 변화
속도라 육성의 정답 자체가
사라지는 중

03

기존 OJT의 한계

OJT(On-the-Job Training) 등 기존
현장 육성 방식이 통하지 않는
상황 증가

04

육성·배치의 복잡화

역할이 유동적일수록 명확한 평가
기준이 없으면 육성과 배치가
어려워짐

대응 **스킬셋 재정의**

역할의 경계를 허무는 일과 개인의 스킬을 정의하는 일을 함께 진행해 강점·약점에 대한 눈높이를 맞춤

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 27쪽

SECTION 4.8

에이전트 시대의 새 직무

FDE·AI Enablement Lead·Agent boss

KEY INSIGHT 범용 에이전트는 이미 누구나 씀 — 차이는 자기 업무 흐름에 맞춘 에이전트를 만들 수 있는지에서 생김

USE

쓰는 사람

Claude Code · ChatGPT 같은 범용 에이전트에
코딩·조사 업무를 위임

이미 일상 업무에서 널리 쓰임

누구나 같은 도구를 같은 방식으로 사용

BUILD

만드는 사람

사내 시스템·데이터에 붙는
업무 전용 에이전트를 직접 구축

이 덱의 대상

AWS 서비스로 업무 에이전트를 만드는 방법

2025년을 에이전트 원년이라 부른 이유

에이전트를 쓰는 경험이 개발자 중심으로 먼저 퍼짐 — 이제 업무에 맞춰 만드는 단계로 넘어감

Source: 미타 미노루(KDDI) 발표(2025-08) 원본 p.5~6

💡 **KEY INSIGHT** 세 조사가 공통으로 지목한 원인은 업무 흐름 통합 부재 — 모델을 바꿔도 현업이 쓰지 않으면 손익 지표는 움직이지 않음

조사	측정 단위	핵심 수치	원인으로 지목한 항목
MIT NANDA The GenAI Divide	기업 생성형 AI 파일럿	95%가 손익 효과 없음	업무 흐름 통합 부재, 쓰면서 학습하는 구조 부재
Gartner 전망	에이전틱 AI 프로젝트	40% 이상이 2027년 말까지 취소	비용 증가, 불분명한 사업 가치, 위험 관리 미흡
BCG The Widening AI Value Gap	기업	대규모 가치 창출 5%, 효과 미미 60%	업무 재설계 없이 도구만 도입

모델 교체·라이선스 구매는 예산만 있으면 집행됨 — 배포한 에이전트를 현업이 매일 쓰게 만드는 사람은 찾고 길러야 해서 시간이 걸림

Source: MIT NANDA, The GenAI Divide(2025.08, 보도 기준) · Gartner(2025.06) · BCG, The Widening AI Value Gap(2025.09)

4.8 AI 정착 직무의 채용 공고 증가

KEY INSIGHT 미국 FDE 공고는 1년에 643건에서 5,330건 — 현업 정착을 사람이 메우는 구조가 채용 수요로 확인됨

+729%

미국 FDE 채용 공고

643건 → 5,330건, 2025.04 → 2026.04

+281%

AI 트레이너 공고

공고 3,500만 건 분석, 2021 → 2026

+196%

업무 자동화 전문가 공고

같은 분석, 5년 누적

읽을 때 주의	내용
기간이 다름	FDE 는 1년, Randstad 는 5년 누적 — 두 증가율을 크기로 견주지 않음
직무 정의	AI 트레이너 공고에는 데이터 라벨링 업무가 섞였을 수 있음

Source: LeadDev, The rise of the forward-deployed engineer(2026) · Randstad Workmonitor 2026(Staffing Industry Analysts 보도) · Microsoft WTI 2025

KEY INSIGHT 공급사 FDE는 첫 구현, 고객사 AI Enablement Lead는 현업 확산, 실무자 Agent boss는 위임·검수를 맡음 — 2층의 소속을 정하는 일이 경영진 몫

층·명칭	소속	책임 범위	성과 기준
1층 FDE(AI 현장 배치 엔지니어)	공급사	고객 업무 문제를 에이전트로 구현하고 운영으로 전환	운영에 올라간 업무 수, 고객 손익 지표 변화
2층 AI Enablement Lead(AI 활용 확산 리드)	고객사	현업의 에이전트 기획·제작·운영 교육과 지원, 공급사 산출물 인수	현업이 직접 만든 에이전트 수, 부서별 실사용률
3층 Agent boss(에이전트 관리자)	모든 실무자	에이전트 제작, 업무 위임, 결과 검수	맡긴 업무의 품질·처리 시간, 검수 누락 건수

2층 배치: 에이전트를 먼저 만들어 본 현업 실무자를 사업 부서 소속으로 두고, 플랫폼·보안 기준은 IT 부서가 함께 지원

Source: MSAP.ai, AI 시대 일하는 방식 백서 5.3절(2026.10) · LeadDev(2026) · Microsoft WTI 2025

4.8 사람 대 에이전트 비율 지표

KEY INSIGHT 부서 경쟁력을 인원수와 운영 에이전트 수의 조합으로 봄 — 결과 지표와 짝을 지어 보고할 때만 성과로 인정함

지표	정의	해석할 때 주의할 점
운영 에이전트 수	지난 분기에 실제 업무 결과를 낸 에이전트	만들어 두기만 한 에이전트는 제외
사람 대 에이전트 비율	운영 에이전트 수 ÷ 부서 인원(분기 말)	비율 자체를 목표로 두지 않음
위임 업무 비중	목록 업무 중 에이전트가 처리하고 사람은 검수만 하는 비율	검수 담당자가 없는 업무는 분모에서 제외

선행 신호	짝으로 보고할 결과 지표	판정
운영 에이전트 수 증가	해결 건수, 처리 시간	결과 개선이 함께 있을 때만 성과
위임 업무 비중 증가	재작업률, 검수 불합격률	불합격률이 오르면 위임 범위 축소
토큰 사용량 증가	결과 단위당 비용	단위 비용이 오르면 원인 점검

Source: Microsoft, 2025 Work Trend Index(2025.04) · MSAP.ai 백서 8.2절 · 헤럴드경제, 포스코DX 1인 N에이전트(2026.06)

SECTION 4.9

암묵지와 학습하는 조직

암묵지를 형식지로 바꾸는 팀

KEY INSIGHT 형식지를 다루는 단계는 AI 지원이 가능하고 공동화는 아직 대체가 어려움 — 암묵지를 주고받는 활동이 줄면 지식 순환이 멈춤

암묵지

형식지

공동화 Socialization · 암묵지 → 암묵지

데일리 스크럼
페어·몹 프로그래밍
사용자 관찰
잡담·대화·공동 경험

AI 대체 어려움

표출화 Externalization · 암묵지 → 형식지

인셉션 덱
사용자 스토리 매핑
프로덕트 백로그 작성
설계

AI 지원 가능

내면화 Internalization · 형식지 → 암묵지

스프린트 리뷰
스프린트 회고
결과를 써 보며 익히기

AI 지원 가능

연결화 Combination · 형식지 → 형식지

스프린트 플래닝
리파인먼트
개발 · 테스트

AI 지원 가능

SECI 모델

암묵지: 경험과 감각으로 아는 지식
형식지: 문서와 코드로 적힌 지식
둘이 서로 바뀌며 커지는 지식 창조 과정

사람이 맡을 영역
현 시점에는 AI 대체가 어려움

AI가 도울 수 있는 영역
형식지를 입력이나 출력으로 다룸

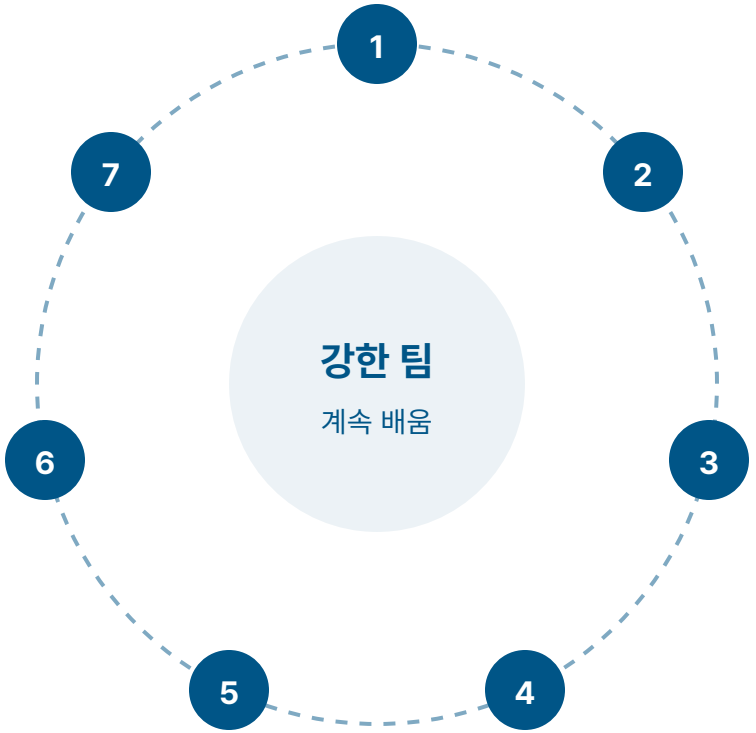
사람끼리 지식을 주고받고
바꾸는 과정이 아웃컴을 만듦

활동 배치는 발표 자료 기준
내면화의 셋째 항목은 보충

Source: 오요베 다카오, Scrum Fest Fukuoka 2026 발표 · SECI 모델 원전(보충): Nonaka & Takeuchi, The Knowledge-Creating Company, 1995

4.9 강한 팀의 일상 학습 루프

KEY INSIGHT 개선 아이디어가 팀 안에서 나오는 상태가 강한 팀의 기준 — 에러를 읽지 않고 AI에 되묻기만 하면 배움이 남지 않음



1	**아침 캐주얼 공유**	어제 나온 AI 업데이트 공유
2	**시도**	AI-DLC·사양 주도 개발 등 실험
3	**새 방식 적용**	효과 있는 기법을 업무에 반영
4	**Toil 줄이기**	반복 작업을 에이전트로 전환
5	**사용자 의견 듣기**	피드백으로 다음 가치 확인
6	**정보 발신**	블로그·사내 공유·행사 발표
7	**회고**	무엇을 그만둘지 스스로 결정

배움이 남지 않는 방식 (발표 자료: 아마 최악의 방식)

AI에 작업 의뢰 — 에러를 읽지 않고 AI에 전달 — 성공까지 반복

코드 내용은 끝까지 보지 않음 — 결과물은 남아도 팀에 남는 지식은 없음

Source: 원 발표 자료 35~41쪽(루프), 41쪽(피할 방식)

하네스 엔지니어링·루프 엔지니어링은 원 발표가 시도 중인 기법으로 이름만 소개함(정의는 원 발표에 없음)

4.9 음성 질문에서 문서까지 4단계 흐름

KEY INSIGHT 말하기만 하면 되므로 진입 장벽이 낮고 질문이 곧 문서 재료가 됨 — 문서화를 위한 시간을 따로 낼 필요가 줄어듦

STEP 1 엔지니어



정리돼 있지 않아 막힌 점을
말로 질문

STEP 2 음성 인식



질문 음성을
텍스트로 변환

STEP 3 AI Agent



텍스트를 해석하고
구조화된 문서를 작성

STEP 4 형식지 자산



문서로 남아
다음 사람이 재사용

왜 음성인가

'말하기만 하면 된다'는 낮은 진입 장벽 덕분에 질문이 바로 문서 재료가 됨

질문 과정에서 필요한 암묵지가 자연스럽게 끌려 나와 형식지로 바뀌고, 학습 자산이 만들어짐

Source: 원 발표 자료(원커리어, 2025.04.23) 14쪽 흐름도를 다시 그림

4.9 변화가 조직에 퍼진 두 경로

KEY INSIGHT 현장의 시도를 의사결정 단계로 끌어올리자 직군 이동과 AI 활용 공유가 연쇄로 일어남 — 다른 팀 사례에 대한 관심이 확산을 이끔

제품을 개선해 온 현장 학습 루프가 조직에서도 같은 구조로 돎

경로 1

직군을 넘는 의욕 전파

계기 자신도 해 보고 싶다는 자발적 의사

엔지니어뿐 아니라 디자이너·마케팅 멤버도
PdM을 해 보고 싶다는 목소리를 냄
그 목소리에 맞춰 직군 이동이 진행되고
반대 방향 이동도 있음

경로 2

AI 활용·업무 개선 확산

계기 개인의 시도를 조직 전체에 공유

멤버가 스스로 시작한 AI 활용·업무 개선이
공유되어 조직 전체로 퍼진 사례 발생
동료의 AI 활용에 자극받아 새로 쓰기
시작한 멤버도 나옴

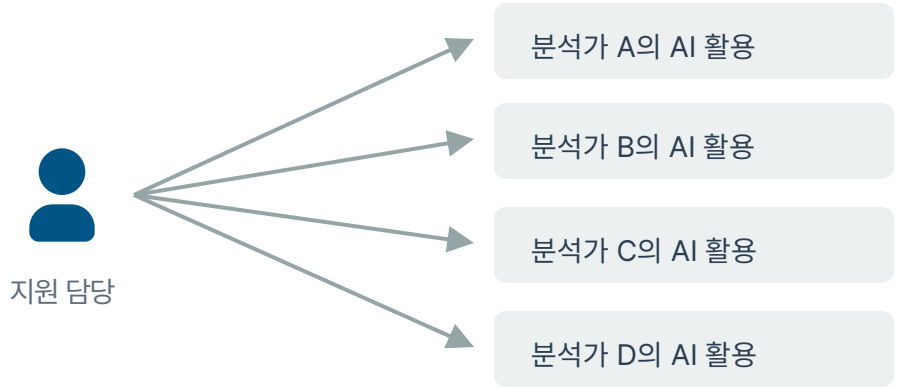
작은 계기가 큰 변화로 이어짐 — 다른 팀과 다른 사례에 대한 관심이 확산을 계속 이끔

Source: 원 발표 다치바나 유야(Recruit), Givery Summit 2026, 2026-07-22 32쪽

4.9 개인별 AI 활용 지원은 조직 역량으로 남지 않음

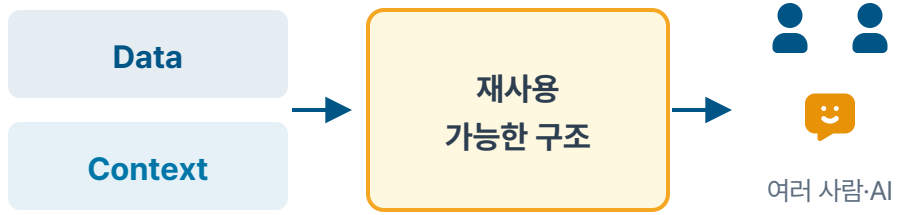
KEY INSIGHT 개별 지원은 효과가 있지만 확장이 어려움 — 여러 사람과 AI가 재사용하는 구조로 옮겨야 효과가 쌓임

이전 방식 — 2025년 하반기, 개인별 AI 활용 지원



- 한 사람씩 보면 효과가 있음
- 사람 수만큼 시간이 들어 확장이 어려움

새 방식 — 재사용 구조에 올리기



- 한 번 정리한 정의와 배경을 여러 명이 재사용
- AI 에이전트도 같은 기준으로 답함
- 지원 담당이 빠져도 구조는 남음

발표자가 데이터 엔지니어로 옮긴 직접적 이유 — 개인 지원을 구조 정비로 바꾸기 위함

Source: hira(jakko), Data Career Talk #2 발표(2026-05-15) · 발표자 블로그 회고 글(원 발표 인용)

KEY INSIGHT 회고 결과를 프로젝트 사이에 이어 보관해야 개인 경험이 조직의 판단 기준이 됨 — 프로젝트마다 버려지면 같은 실패가 반복됨

KPT 보드 예시

발표 자료 화면 구성을 다시 그림

Keep 잘된 점 · 계속할 일	Problem 문제점 · 과제	Try 다음에 시도할 개선
매일 짧은 진행 공유	배포가 느림	모니터링 보강
AI 산출물 검토 체크리스트	AI 초안이 검토 없이 공유됨	자동 테스트 확대
합의 내용 즉시 기록	요구 변경 이력 누락	승인자 지정 규칙 도입

운영 방법 (보충)

1. **주기**
   스프린트나 단계가 끝날 때 1회
2. **기록 위치**
   프로젝트 간 공유되는 위키 한곳
3. **다시 보기**
   다음 프로젝트 착수 회의에서 Try 확인
4. **AI 활용**
   회고 기록 요약은 AI, 패턴 판단은 사람

Source: 원 발표(i3DESIGN, AI 구동 개발 스터디 제7회) 시간 투자 쪽 KPT 화면

보드 항목은 원 발표 화면의 예시를 참고해 다시 쓴 것이고 운영 방법은 보충

SECTION 4.10

손익으로 재는 AI 성과

과정 보고를 손익 보고로 바꾸는 법

KEY INSIGHT 외부 조사는 모두 손익으로 성과를 판정함 — 내부 보고가 진행률·사용량에 머물면 성공으로 보고한 과제가 경영 판단에서 효과 없음으로 분류됨

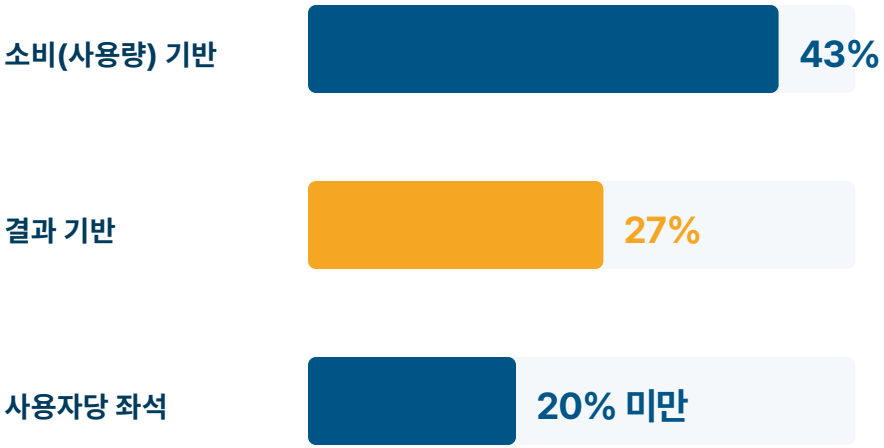
조사	성과 판정 기준	기준을 넘은 비율
BCG The Widening AI Value Gap	매출 성장·비용 절감에 나타난 가치	대규모 가치 창출 5%
MIT NANDA The GenAI Divide	파일럿의 측정 가능한 손익 효과	손익 효과 없음 95%
McKinsey The State of AI 2025	AI에 귀속되는 EBIT 기여	고성과 기업 약 6%

항목	과정 보고(기존)	결과 보고(전환 후, 가상 예시)
산출	에이전트 4개 제작, 블로그 60편 생성	블로그 경유 문의 18건 중 영업 기회 전환 5건
사용	사용자 120명, 토큰 사용량 2배	상담 1건당 처리 시간 12분 → 5분
비용	라이선스 비용 집행률 80%	외주 제작비 월 절감액, 재배치한 인력 시간

Source: BCG(2025.09) · MIT NANDA(2025.08) · McKinsey, The State of AI 2025(2025.11) · 하단 표는 MSAP.ai 백서 7.1.2절 가상 예시

4.10 결과 기반 과금의 등장과 과금 선호

KEY INSIGHT 공급사는 해결 1건 단위로 값을 받기 시작했고 좌석 과금 선호는 20% 미만 — 계약의 핵심 조건은 단가보다 해결 정의와 측정 주체



Intercom Fin 결과 1건 0.99달러(2026.10 공개 가격표)

계약 점검 항목	확인할 내용
해결 정의	무엇을 1건으로 세는가(고객 확인, 재문의 없음 기간)
측정 주체	건별 대화 기록 열람권과 월간 대사 절차가 있는가
하한 보장	최소 약정 건수·금액과 미달 시 이월 조건
예외 처리	사람에게 인계된 건과 재문의 건을 어떻게 세는가
품질 연동	만족도 기준 미달 건을 과금에서 빼는가

Source: Futurum Research, 1H 2026 Enterprise Software Decision Makers(2026.05) · Intercom Fin 가격표(2026.10) · Zendesk(2024.08)

4.10 에이전트 결과 지표의 3단 정의

KEY INSIGHT 목적 → 결과 단위 → 손익 연결로 정의하면 에이전트 성과를 재무 부서와 같은 숫자로 설명함 — 첫 분기 추적 지표는 세 개로 제한

업무	결과 단위	단가 산정 근거(가정)	월 건수	월 손익 기여
고객 상담	해결 1건	상담원 1건 처리 비용 8천 원 – 에이전트 과금 1천 3백 원	6,000건	비용 절감 약 4,020만 원
마케팅	영업 기회 전환 리드 1건	리드 1건 확보 비용 15만 원(광고 기준)	40건	비용 절감 600만 원
제안 문서	검수 통과 문서 1편	외주 작성비 1편 80만 원	6편	비용 절감 480만 원

1 시스템이 이미 세는 지표

건수를 따로 세는 수고가 없음

2 손익계산서 한 줄과 바로 연결

인건비·외주비·광고비 줄에 대응

3 업무 전문가가 판정 기준을 씀

합격 기준을 현업이 직접 정함

Source: MSAP.ai, AI 시대 일하는 방식 백서 7.4절(2026.10) — 단가·건수는 설명용 가정값

4.10 생성형 AI 활용 수준별 경쟁력 격차

KEY INSIGHT 격차는 사업 가치·생산성·비용·전환 속도에서 동시에 벌어짐 — 사업 개발과 생산성 개선을 함께 추진해야 함

	사업 가치 확보	조직 생산성	비용 구조	전환 속도
적극 활용한 기업	생성형 AI로 새로 만들 수 있는 사업 가치를 누가 먼저 가져가는지	**2~10배** 정도의 차이 (발표 자료자 추정, 미확인)	비용 구조 차이가 제품 가격·품질 경쟁력의 차이로 이어짐	생성형 AI를 전제로 한 조직 체계로 옮겨 가는 속도의 차이
활용에 소극적인 기업				

기업이 함께 다뤄야 할 두 주제

주제 1 · 발표 자료가 집중한 주제

생성형 AI 기반 사업·제품 개발

생성형 AI가 있어야 성립하는 새 서비스 만들기

×

주제 2

생성형 AI 활용 생산성 개선

기존 업무에 생성형 AI를 넣어 철저히 효율화

Source: 원 발표 22~24쪽(가지타니 겐토). 생산성 배수는 원 발표자 추정

4.10 AI 적용 지점 판정 예시

KEY INSIGHT AI 효과를 작업 비중으로 환산해 전체 단축률을 미리 계산함 — 적용 순서를 근거 있는 숫자로 정할 수 있음

공정	AI로 줄일 작업 (선택)	전체 대비 비중
브리프 작성	**시장·경쟁 분석**	**5.43%**
브리프 작성	**계획·조율**	**2.33%**
브리프 작성	**문제 정의**	**9.30%**
브리프 작성	**프로토타이핑**	**3.88%**
브리프 작성	**사용자 조사 분석**	**3.10%**
브리프 작성	**사용자 조사 준비**	**5.43%**
PRD 작성	**리뷰·협업**	**3.10%**
선택 작업 7개 합계		**32.57%**

발표 자료 표는 공정·작업별 비중(Weight)·영향 점수를 모두 나열하고 AI 대상 작업을 강조함.
사용자 조사 실행, 요건 정의, 릴리스 계획 같은 작업은 대상에서 제외됨.

효과 환산

가정

AI로 선택 작업을 40% 단축

32.57% × 40% ≈ 13%

결과

약 13% 단축

제품 개발 프로세스 전체 시간 기준

발표 자료: 40% 가정과 13% 결과를 제시
보충: 강조 작업 비중 합으로 검산하면
32.57% × 0.4 = 13.0%로 일치

Source: Yoshi Takeda (freee 상무집행임원 VPoE), Findy 주최 행사 발표 자료 (2025.11.13)

작업별 비중·40% 가정·13% 결과는 원 발표 기준(freee 내부 분석, 외부 미확인). 작업 이름은 원본 영문을 한국어로 표기

SECTION 05

개인이 일하는 방식

셀프 매니지먼트·성장 방향·판단력·학습 습관·커리어

SECTION 5.1

AI 시대 셀프 매니지먼트

변화 압박과 FOMO 를 다루는 자기 관리

5.1 AI 시대에 셀프 매니지먼트가 더 중요해짐

KEY INSIGHT 지시대로 반복하는 매뉴얼 업무는 AI로 대체됨 — 남는 지식 노동을 받치는 역량이 셀프 매니지먼트임

구분	매뉴얼 워커	지식 노동자(Knowledge Worker)
일하는 방식	지시와 절차를 따라 작업함	스스로 생각해 성과를 냄
업무 성격	같은 작업을 반복함	매번 다른, 재현되지 않는 일을 함
효과적인 동기	외적 동기(보상·지시)가 효과적	내적 동기(재미·의미)가 효과적
AI 시대의 변화	**기계와 AI로 대체됨**	**사람의 감각과 개성을 살리는 일로 남음**

결론 **지식 노동을 받치는 역량이 셀프 매니지먼트임**

Source: 구라누키 요시히토(소닉가든), Scrum Fest Sendai 2026 기조연설(2026.7.10)

보충: 지식 노동자는 피터 드러커가 널리 알린 개념

5.1 변화 압박과 타율적 행동

KEY INSIGHT 조급함이 커지면 해야 하는 일이 우선됨 — 외부 기준이 모르는 사이 일의 방향을 정함

가속하는 변화 주기가 심리적 압박을 거쳐 타율적 행동으로 이어지는 흐름

① 환경 변화

- 기술 발전 주기의 단축
- 시장·고객 요구의 변화
- 경쟁과 도태의 속도 상승

개인이 통제하기 어려운 외부 요인

→

② 심리적 압박

- '따라가야 한다'는 조급함
- '뒤처진다'는 두려움(FOMO)

외부 요인이 감정으로 바뀌는 단계

→

③ 타율적 행동

- '해야 한다'가 먼저 결정됨
- 모르는 사이 외부가 정한 기준으로 움직임

일의 방향을 스스로 정하지 못하는 상태

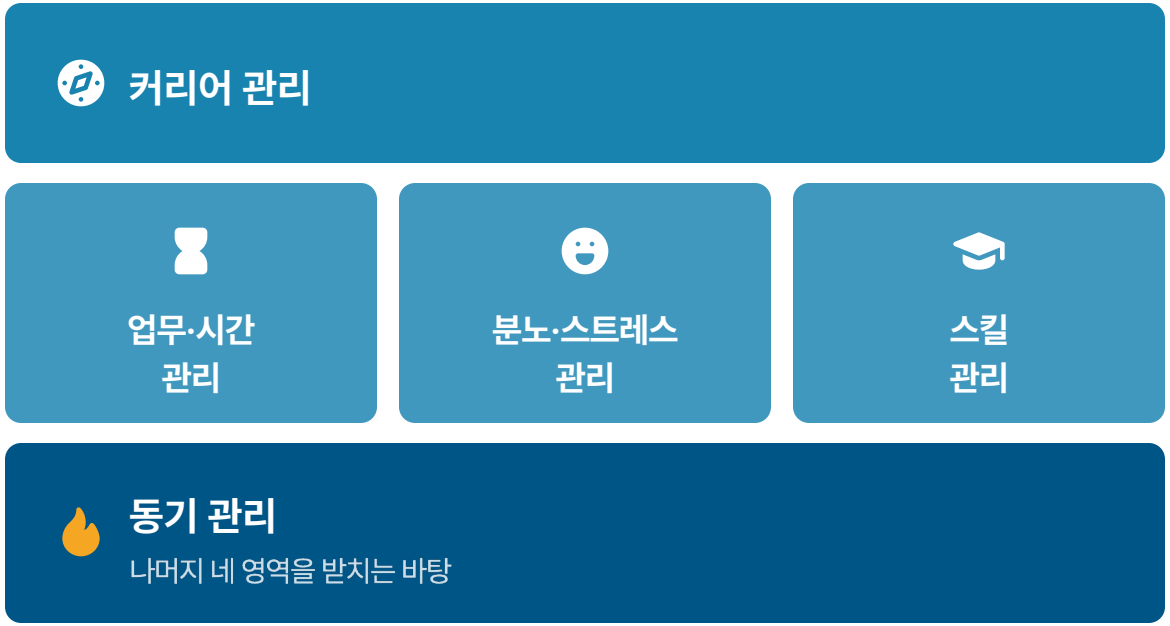
변화 속에서 자율성을 지키기는 쉽지 않음 — 압박이 행동을 정하기 전에 그 흐름을 알아차리는 것이 출발점

Source: 원 발표 4쪽(변화 속에서 자율성을 지키기 어려운 이유)

5.1 자기 관리 다섯 영역의 구조

KEY INSIGHT 동기 관리가 나머지 네 영역의 바탕임 — 동기가 흔들리면 시간·스트레스·스킬·커리어 관리도 함께 흔들림

발표 자료의 구조도 — 네 영역이 동기 관리 위에 놓임



바탕이 흔들리면 위의 네 영역도 함께 흔들림

영역별로 다루는 질문

동기	무엇을 왜 하고 싶은가
업무·시간	무엇을 하고 무엇을 하지 않을까
스트레스	불안과 조급함에 어떻게 대응할까
스킬	무엇을 키우고 무엇을 AI에 맡길까
커리어	어떤 엔지니어가 되고 싶은가

영역별 질문은 발표 자료 각 장의 요지를 한 줄로 줄인 것
스트레스 영역의 발표 자료 명칭은 분노·스트레스 관리

Source: 원 발표 7쪽·20쪽(셀프 매니지먼트의 구조)

5.1 AI로 줄어든 시간은 새 업무로 채워짐

KEY INSIGHT 효율이 오를수록 일의 총량이 늘어나는 제번스 역설이 업무에도 나타남 — 생긴 여유를 어디에 쓸지 스스로 정해야 함

AI 도입 뒤 작업 시간 (발표 자료자 경험)

출시 전 테스트 항목 분류(트리아지)

팀 횡단 보고서 작성



막대 길이는 개념 비교용

작업에 쓰는 시간은 줄고
생각에 쓰는 시간이 크게 늘어남

제번스 역설(Jevons Paradox)

자원 이용 효율이 오를수록 자원 사용량이 오히려 늘어남



AI는 일을 없애기보다 일을 늘리고 있음
생긴 여유를 어디에 쓸지는 스스로 정해야 함

보충

1865년 영국 경제학자 윌리엄 스탠리 제번스가
석탄 이용 효율과 소비량 증가를 두고 처음 지적

2025년 저비용 AI 모델 등장 뒤 다시 주목받음
AI 수요 전망 논의에서 자주 인용됨

Source: 원 발표 12~15·19쪽, Northeastern University News(2025.02.07) Jevons paradox and AI

KEY INSIGHT 할 수 있어서 하는 일이 늘어남 — 실행 비용이 낮아진 지금은 필요 여부가 선택 기준임

YAGNI

You Aren't Gonna Need It

원래 뜻 (보충)

나중에 필요할 것 같은 기능을 미리
만들지 않는 XP(익스트림 프로그래밍) 원칙

발표 자료의 적용

AI로 실행 장벽이 크게 낮아짐
→ 일 자체에도 YAGNI 적용이 필요한 시점

그 일, 정말 필요한가

일을 시작하기 전에 던지는 한 가지 질문

할 일을 거르는 흐름

할 수 있는 일

AI 덕분에 손쉽게 만들 수 있는 기능·문서·실험이 계속 늘어남

↓

필요한가?

지금 이 일이 목표에 필요한지로 거름 — 가능 여부는 기준에서 뺌

↓

필요해서 하는 일

집중해서 끝까지 진행

할 수 있어서 하는 일

지금은 하지 않음으로 결정

Source: 원 발표 12쪽 · 보충: Martin Fowler, bliki 'Yagni'(2015)

5.1 그만둘 일을 가리는 세 가지 질문

KEY INSIGHT 세 질문에 분명한 답이 없는 일은 한 번 멈춰 볼 대상임 — 무엇을 버릴지 고르는 일이 가장 큰 여유를 만듦

'다 중요한 일'이라고 느낄 때, 일마다 아래 세 질문에 답해 봄

Q1 얻는 것

지금 이 일을 하면

무엇을 얻는가

Q2 잃는 것

지금 이 일을 하지 않으면

무엇을 잃는가

Q3 방치 결과

이 일을 그대로 두면

어떻게 되는가

세 질문 모두 답이 분명하지 않으면 일단 멈춰 봄

멈춘 뒤 문제가 생기면 다시 시작하면 됨 — 고르고 버리는 결정이 가장 큰 여유를 만듦

Source: 원 발표 29~30쪽(2장 업무·시간 관리)

SECTION 5.2

수직적 발달과 추상 역량

수평적 성장보다 수직적 발달

KEY INSIGHT 지식이 늘어도 생각의 틀이 그대로면 일하는 방식은 바뀌지 않음 — 두 축은 서로 다른 방법으로 자람

수평적 발달

양적 성장

지식과 스킬을 얻는 성장

- 개별 능력의 종류가 늘어남
- 새 언어, 새 프레임워크, 도메인 지식
- 무엇을 얼마나 아는지로 측정 가능

+

수직적 발달

질적 성장

생각의 틀이 바뀌는 성장

- 사물을 보는 관점과 판단 기준이 바뀜
- 사람으로서 그릇이 커짐
- 같은 지식으로도 다른 결정을 내림

발표 자료의 비유

수직적 발달 → OS에 해당 (생각의 틀·가치관)

수평적 발달 → 애플리케이션에 해당 (직무별 지식·스킬)

Source: ONE CAREER Inc., Forkwell 행사 발표(2026-04-24). 수평적·수직적 발달 구분은 성인 발달 이론 계열의 개념

5.2 생성형 AI 확산으로 수직적 발달이 중요해짐

KEY INSIGHT 지식 보유의 가치는 낮아짐 — AI가 내놓는 지식을 골라 쓰는 생각의 틀이 성과를 가름

근거: 생성형 AI가 인터넷의 지식 대부분을 이미 학습함 → 지식량만으로는 차별화가 약해짐

수평적 발달

AI가 지원 가능

지식과 스킬을 얻는 양적 성장

- 모르는 문법·API는 AI에게 바로 물어봄
- 처음 보는 기술도 AI 설명으로 빠르게 따라잡음
- 개인의 지식량 차이가 좁혀짐

수직적 발달

중요도 상승

생각의 틀이 바뀌는 질적 성장

- AI가 내놓은 많은 답 가운데 무엇을 쓸지 판단
- 답이 맞는지, 써도 되는지 검토
- AI가 대신하기 어려운 영역으로 남음

AI가 주는 대량의 지식을 살리려면 그 지식을 다루는 생각의 틀부터 자라야 함

Source: ONE CAREER Inc., Forkwell 행사 발표(2026-04-24). 카드 아래 세부 항목은 원 발표 요지를 실무 예로 풀어 씀

KEY INSIGHT AI가 지시를 충실히 따르는 단계에 이미 가까움 — 사람에게 자기 주도형 이상의 역할이 더 일찍 요구됨



발표 자료의 해석

AI는 이미 1단계에 가까움

지시를 받아 충실히 수행하는 역할은
AI가 상당 부분 맡을 수 있음

사람은 2·3단계를 더 일찍 요구받음

스스로 기준을 세우고 과제를 정하는
자율적 행동이 필요한 시점이 앞당겨짐

용어

성인 발달 이론: 어른이 된 뒤에도 지성과 의식이 단계적으로 자란다는 심리학 이론
영문 단계명: Socialized · Self-authoring · Self-transforming Mind (보충)

Source: ONE CAREER Inc. 발표(2026-04-24) — Robert Kegan 성인 발달 이론, Kegan & Lahey, Immunity to Change(Harvard Business Press, 2009)의 도식을 참고해 다시 그림

5.2 학습과 사고 훈련을 함께 진행

KEY INSIGHT 무엇을 시킬지와 실행 여부는 당분간 사람이 정함 — 이 판단의 질이 AI 활용 품질을 정함

AI에게 많은 일을 맡겨도 당분간 사람이 정하는 두 가지

- 무엇을 시킬지 정하기 (과제 설정)
- 그 일을 해도 되는지 결정하기

학습 (수평적 발달)

- 기술 스택 확장
- 새 언어와 프로그래밍 패러다임 습득
- 비즈니스·도메인 지식 심화

AI 설명과 실습으로 속도를 높일 수 있는 영역

사고 훈련 (수직적 발달)

- 시야를 개인에서 팀·조직 단위로 넓힘
- 주어진 과제 해결에서 과제 정의로 이동
- 학습과 같은 기간에 병행

AI에게 맡기기 어려운 판단을 기르는 영역

Source: ONE CAREER Inc., Forkwell 행사 발표(2026-04-24)

5.2 구체적 스킬일수록 빨리 쓸모가 줄어듦

KEY INSIGHT AI 활용 스킬도 특정 기술에 묶인 스킬이라 계속 갱신해야 함 — 오래 쓰이는 쪽은 검증·책임·소통 같은 추상 역량임

구체 — 특정 기술에 묶인 스킬 → 추상 — 기술에 묶이지 않는 스킬

AI에 넘길 스킬

기술 세부 지식
정형 코드 작성
검색

이미 쓸모가 줄어든 쪽

새로 필요한 스킬

AI 활용 범위 확대
풀스택화
변화 속도 대응

AI 활용 스킬도 곧 갱신 대상

변하지 않는 역량

코드 읽기와 출력 검증
결과 책임, 대인 소통
경험의 추상화

오래 쓰이는 쪽

구분	유지 기간	해당 스킬
특정 기술에 대한 스킬	짧음	넘길 스킬, 새로 필요한 스킬 대부분
특정 기술에 묶이지 않는 스킬	김	변하지 않는 역량

발표 요지: 추상 스킬일수록 오래 쓰이고 구체 스킬일수록 빨리 쓸모가 줄어드는 법칙은 AI 이전부터 있었음

Source: 원 발표(스리세이크, 2026.02.04) 여담: 스킬의 추상도와 유효 기간 쪽

원 발표는 식품의 유통기한에 빗대 설명함. 세 구분의 축 배치는 보충

5.2 개별 기술 세부 지식의 가치 감소

KEY INSIGHT 세부 사항은 AI에 물어 바로 확인할 수 있음 — 다만 비공개 구현과 AI 답의 정확성은 사람이 직접 확인해야 함

달라진 점

AI에 물으면 대부분 알 수 있음

라이브러리·언어·서비스의 세부 동작

미리 외워 두는 지식의 가치는 줄어듦

공개된 구현은 꽤 정확히 답함

코드를 AI에 읽히면 동작 근거까지 짚음

오픈소스 라이브러리 조사에 특히 유효

AI 만으로 끝나지 않는 경우

구현이 비공개라 직접 돌려 봐야 아는 동작

구현이 공개돼 있어도 AI 출력은 검증 필요

그래도 바로 꺼낼 지식은 신뢰로 이어짐

고객 요청 예시

〇〇 를 구현할 방법이 있는지 문의

1단계 먼저 목표로 삼을 수준

다음 회의까지 조사해 제안하기로 답함

이 대응으로도 문제없음

2단계 신뢰도가 올라가는 대응

그 자리에서 쓸 만한 기술 ××를 짚고

상세 안은 다음 회의에 제안하기로 답함

Source: 원 발표(스리셰이크, 2026.02.04) 개별 기술의 세부 지식 쪽

고객 대화는 원 발표 예시의 요지. 회색 보조 문구는 보충

SECTION 5.3

AI 의존과 판단력

AI 의존이 만드는 스킬 공백과 결과 책임

5.3 AI 의존과 판단력 저하 위험

KEY INSIGHT AI 결과를 평가하지 않고 받기만 하면 평가 능력 자체가 줄어듦 — 반박하지 않는 AI와 오래 일할수록 근거 없는 확신이 커짐

사고의 세 단계와 AI가 끼어드는 자리

3 사고의 외부화(AI)

AI에 생각 자체를 외주화
1·2 단계를 대신해 판단력이 비게 되는 위험

2 숙고적 사고

시간을 들여 깊이 따져 보는 사고
반복하면 1 단계 반응으로 굳어짐

1 직관적 사고

빠르고 무의식적인 판단
경험으로 자동화된 반응

인용 요지 — Ethan Mollick

일을 넘길 때마다 그 일을 평가할
전문성을 키울 기회를 잃음

결과는 얻지만 우리의 자리는 직접 하는 사람·조수에서
지켜보는 청중으로 바뀜

"On Working with Wizards", One Useful Thing

반박하지 않는 AI

AI는 내 아이디어를 거의 부정하지 않음
내 생각을 되돌려 보여 주는 대화에 가까움

이런 대화가 10년 쌓이면 근거 없는 확신이 생김
→ 스스로 검증하는 겸손한 태도가 필요함

Source: 원 발표 p.20–22 · Ethan Mollick, "On Working with Wizards"(oneusefulthing.org)

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

5.3 AI 의존이 만드는 스킬 공백

KEY INSIGHT 배우지 않아도 결과가 나오는 환경에서는 스킬이 비어 감 — 원리를 묻는 습관이 공백을 일찍 드러냄

일어나는 일

그럴듯하게 동작하는 결과물

코딩 경험이 없어도 애플리케이션을
만들 수 있고, 꽤 그럴듯하게 동작함

→ 모른다는 사실을 알아차리기까지 오래 걸림

학습 없이 돌아가는 업무

AI가 손을 대신 움직일수록 배우지 않아도
일이 진행되고 모르는 사이 스킬이 비어 감

→ 기술 노후화 속도에도 뒤처짐

필요한 자세 — 의도해서 관리하는 스킬

· '이게 왜 동작하지?'라는 질문을 계속 던지고 답을 내 것으로 만듦

· 무엇을 키우고 무엇을 AI에 맡길지 운에 맡기지 않고 스스로 고르며 학습을 계속 갱신함

Source: 원 발표 16~17쪽(AI 시대의 스킬 관리)

5.3 AI 사용 여부와 무관한 결과 책임

KEY INSIGHT 평가는 결과의 품질과 속도로 정해짐 — AI를 쓰지 않으면 불리하고, 썼다면 설명과 사후 대응까지 본인이 맡음

사회인은 과정보다 결과로 평가받음 — AI를 썼는지보다 프로로서 품질과 속도를 냈는지가 기준

감점으로 이어지는 결과

빠르지만 품질이 낮은 결과

AI로 1시간 만에 끝냈지만
품질이 기대에 못 미침

근거를 AI에 돌리는 답변

AI의 답만 근거로 내세우고
본인은 설명하지 못하는 답변

반대로 AI 없이 일하는 것도 큰 불리함

AI를 쓰고 결과까지 책임지는 모습

높은 품질

속도와 함께 결과물의 완성도를 지킴

설명할 수 있음

왜 이렇게 만들었는지 본인 말로 설명

끝까지 대응

문제가 생기면 마지막까지 책임지고 처리

Source: 원 발표(스리세이크, 2026.02.04) 결과에 책임지기 쪽

감점 사례 두 가지와 기대 행동 세 가지는 원 발표 기준. 카드 보조 설명은 보충

5.3 AI 결과의 책임은 쓰기로 정한 사람에게 있음

KEY INSIGHT AI를 사람처럼 대하면 불안이나 과잉 기대가 생김 — 도구로 보면 어디에 어떻게 쓸지를 사람이 설계해야 함이 분명해짐

발표 자료의 예: AI로 움직이는 가위

미용실 대신 AI 가위에 머리 손질을 전부 맡긴 경우

1. **모든 판단을 AI에 맡김**
   원하는 모양을 정하지 않고 알아서 잘라 달라고 함
2. **원치 않는 결과가 나옴**
   예상과 전혀 다른 머리 모양이 됨
3. **가위 탓으로 끝나지 않음**
   도구에는 책임 능력이 없음

테스트로 옮기면: AI에 관점 도출을 맡겨 누락이 생겨도 책임은 맡긴 쪽에 있음

AI를 사람처럼 대할 때

사람과 같은 일을 한다고 여겨 내 일을 뺏긴다는
불안이나 반대로 과잉 기대가 생김

AI를 도구로 볼 때

무엇에 어떻게 적용할지를 사람이 설계함
결과를 확인하고 받아들일지도 사람이 정함

결과와 책임의 주체

AI를 써서 얻은 결과와 책임은 모두
그 도구를 쓰기로 결정한 사람에게 있음

Source: 원 발표(MIXI, 2026.07.23) AI 시대 사고방식의 재정의 쪽

가위 예는 원 발표의 비유를 요지로 정리함. 테스트 업무에 대응시킨 설명은 보충

5.3 AI 시대에도 필요한 기초 지식

KEY INSIGHT 세 주체의 능력 범위를 알아야 일을 맞는 쪽에 맡길 수 있음 — 로직·AI·사람 사이의 작업 배분이 품질과 비용을 결정함

질문 1
컴퓨터는 무엇을 할 수 있는가

정확한 계산과 로직 처리

같은 입력에 같은 결과

맡길 일

규칙이 정해진 처리·계산

질문 2
AI는 무엇을 할 수 있는가

생성 · 요약 · 제안

약점 다섯 가지를 함께 인지

할루시네이션 · 최신 정보 · 논리 · 역방향 추론 · 비용

맡길 일

초안 작성·코드 제안·요약

질문 3
사람은 무엇을 할 수 있는가

과제 설정과 요건 정의

기존 자료에 없는 발상

중장기 판단과 책임

맡길 일

결과 검증과 최종 결정

Source: 기시다 나오키(LINE야후), GDG Kwansai 발표 자료, 2025-01-18 (세 질문은 원 발표). 각 주체의 능력과 맡길 일 정리는 보충

5.3 검색 능력보다 출처 지정과 검증이 중요해짐

KEY INSIGHT AI에 문서·이슈·구현 중 어디를 보게 할지 정해야 답의 정확도가 올라감 — 사실 확인의 비중은 오히려 커짐

구글 검색 능력

예전만큼 차별점이 되지 않음

대량 정보 처리는 AI가 더 잘함

→

어디를 찾게 할지 지정

아직은 사람이 알고 있어야 함

출처에 따라 답의 정확도가 갈림

→

사실 확인

AI가 찾은 답의 근거 확인

중요성이 오히려 커짐

라이브러리 동작을 조사할 때 AI에 보게 할 출처

인터넷 글

블로그·Q&A 사례

버전이 다를 수 있어

단독 근거로 부족함

공식 문서

의도된 사용법과

지원 범위 확인

이슈

알려진 버그와

우회 방법 확인

구현 코드

실제 동작의

최종 근거

발표 요지: 인터넷 글 외에 문서·이슈·구현까지 보게 해 탐색 위치를 정해 줌

Source: 원 발표(스리세이크, 2026.02.04) 검색하기 쪽

출처별 용도 설명은 보충

SECTION 5.4

변화 속도에 맞춘 학습 습관

짧은 학습 주기와 발신 습관

KEY INSIGHT 하루 5~10분 단위의 반복이 성장을 만듦 — 큰 시간을 따로 내지 않아도 오늘 바로 시작 가능함

작게 배우고, 배우기 쉬운 환경을 만들고, 회고로 굳힘

짧은 학습

마이크로러닝

5~10분 정도의 가벼운 학습을
매일 습관으로 넣음

예(보충): 출근 뒤 공식 문서 한 절 읽기

학습 환경

학습용 채널

사내 Slack에 학습용 채널을 만들어
배운 것을 가볍게 공유함

예(보충): 읽은 글 링크와 한 줄 소감

회고

YWT 일일 보고

일일 보고에 세 줄로 짧게 정리함

- **Y** 한 일
- **W** 알게 된 일
- **T** 다음에 할 일

일본 기업에서 널리 쓰는 회고 형식(보충)

Source: 원 발표 18쪽(스킬을 쌓아 다듬기)

KEY INSIGHT 개발 방식의 기준이 몇 달 단위로 바뀜 — 지난해 방식을 그대로 이어 쓰기 어려움

그때 어떻게 개발했는지 떠올려 보면 — 기준이 바뀌는 간격이 점점 짧아짐



베스트 프랙티스가 순식간에 뒤집히는 시기

지금까지 없던 속도의 변화 속에서 무엇을 따라가고 무엇을 흘려보낼지 스스로 정하는 일이 필요해짐

Source: 원 발표 3쪽(변화의 속도에 대한 질문)

KEY INSIGHT 알고리즘·네트워크·소프트웨어 공학 지식이 있어야 AI에 맡길 수 있는 일이 늘어남 — 모르는 영역은 AI 결과도 검증하지 못함

준비 1 · 공부

AI로 안전한 소프트웨어를 만들려면 확실한 지식과 경험이 필요함

모르는 영역은 AI가 낸 결과도
맞는지 판단할 수 없음

1

모르는 것을 정말 아는 것으로 바꾸는 능력

아는 것 같은 상태에서 멈추지 않고, 설명할 수 있을 때까지 파고듦

2

흥미를 가진 주제를 깊이 파고드는 습관

결과보다 깊이 파고드는 과정 자체를 즐기기

3

알고리즘·네트워크·소프트웨어 공학 지식

기초 지식이 있을수록 AI에 맡길 수 있는 일의 선택지가 늘어남

Source: sushichan044, Giiku CAMP 아카데미아 발표(2026-07-08)

5.4 AI 활용 경험 발신과 CLAUDE.md 정리

KEY INSIGHT 새 영역은 경력 차이가 작아 신입도 조직에 기여할 수 있음 — 발신과 지침 정리가 본인 성장과 AI 품질을 함께 올림

AI에 대해 발신하기

신입에게 유리한 이유

새 영역은 신입이 선두 주자가 되기 쉬움

경력자도 공부하지만 혼자서는 다 따라가지 못함

발신하는 자리

팀 안	사내	사외(블로그·발표)

신입이라도 조직과 업계에 기여할 수 있음

변화가 빨라 발신할 소재가 계속 생김

CLAUDE.md 같은 지침 파일 정리

CLAUDE.md: Claude Code가 세션마다 읽는
프로젝트 지침 파일 (보충)

경력자

감각으로 굳은 지식이 많음

전부 글로 옮기기 어려움

신입

배운 점을 그때그때 지침 파일에 적으면

본인의 성장과 AI의 성장이 함께 진행됨

강제 설정은 아니고 매 세션 읽는 맥락으로 동작

Source: 원 발표(스리세이크, 2026.02.04) AI에 대해 발신하기, AI를 키우기 쪽; Claude Code 문서 Memory (code.claude.com/docs/en/memory)

두 쪽을 한 장으로 합침. CLAUDE.md 정의와 동작 설명은 보충

5.4 개발 밖으로 넓어진 AI 활용 범위

KEY INSIGHT AI는 학습·글쓰기·대인 업무에도 쓸 수 있음 — 개발에만 쓰면 얻을 수 있는 시간의 일부만 얻음

개발

지금 가장 많이 쓰는 영역

조사, 설계, 구현, 테스트, 리뷰

공정 전 단계에서 초안과 1차 검토를 맡김

학습

새 지식을 받아들이는 일

요약, 해설, 체계화, 학습 자료 작성

낯선 기술의 개요를 빠르게 잡음

글쓰기

배운 것을 밖으로 내는 일

블로그 글 작성, 발표 자료 작성

구성안·초안 작성 시간을 줄임

대인 업무

소프트 스킬을 보조하는 일

메시지 문안 작성, 상황별 처신 조언

보내기 전 표현과 순서를 점검

Source: 원 발표(스리세이크, 2026.02.04) AI를 활용하기 쪽 (개발·인풋·아웃풋·소프트 스킬 네 영역)

영역 이름은 원 발표의 인풋·아웃풋을 학습·글쓰기로 옮김. 카드의 회색 문구는 보충

5.4 AI로 생긴 시간의 투자처

KEY INSIGHT 줄어든 작성 시간을 더 많은 산출물로 채우면 실력 상한은 그대로임 — 고객 대화·강점 연마·사례 축적에 써야 다음 판단의 질이 오름

AI로 줄어든 작성·정리 시간 → **산출물을 더 늘리는 데 쓰지 않고 기초 역량에 투자**

고객·사용자와 직접 대화

제품 이용 현황을 파악
과제를 특정하고
해결 아이디어를 수집

왜 필요한가 (보충)
현장 맥락이 있어야 AI 산출물의
맞고 틀림을 가려냄

자신의 강점 연마

엔지니어링 · 디자인
비즈니스 · 프로젝트 관리 중
강점 분야를 더 깊게

왜 필요한가 (보충)
분야 지식이 깊을수록 틀린
출력을 빨리 찾아냄

실패와 사례로 패턴 축적

실천을 반복하고
실패·성공 사례를 분석해
다음 프로젝트에 반영

왜 필요한가 (보충)
쌓인 패턴이 판단 기준의
근거가 됨

Source: 원 발표(i3DESIGN, AI 구동 개발 스터디 제7회) 시간 투자 쪽

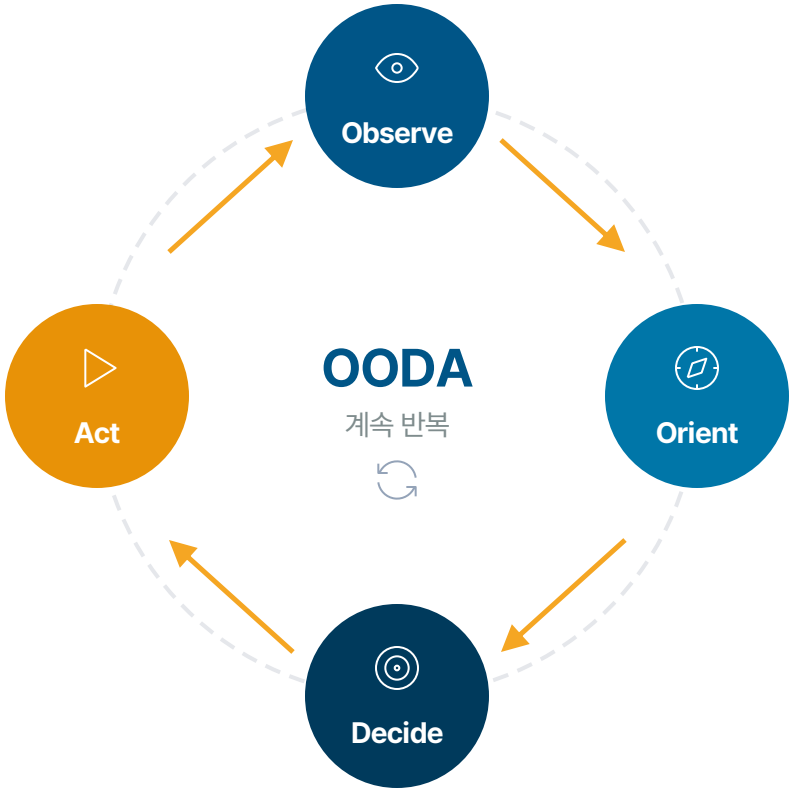
원 발표는 이를 급할수록 돌아가라는 속담으로 요약함. 카드 아래 '왜 필요한가'는 보충

SECTION 5.5

커리어 운영

OODA 루프로 운영하는 커리어

KEY INSIGHT 정답을 미리 정하지 않고 관찰·판단·결정·실행을 반복함 — 변화가 빠른 시장에서는 계획보다 수정 주기가 중요함



Observe 관찰

돈을 내는 사람이 무엇에 기꺼이 지불하는지 봄

고객 · 경영진 · 결재권자

Orient 상황 판단

내 일이 "머릿수" 쪽인지 "사업 기여" 쪽인지 판단

정형 업무 비중 · 매출과의 거리

Decide 결정

먼저 현재 직장에서 할 수 있는 일부터 정함

옮긴다면 리퍼럴 채용 경로 우선

Act 실행

작게 움직이고 여러 커뮤니티에 참여함

실행 결과를 다시 관찰 단계로 넘김

Source: 히사마쓰 쓰요시, AGI 후쿠오카 제12회 발표(2026-10-07) · OODA 루프 개념: 미 공군 존 보이드(John Boyd), Wikipedia "OODA loop"

보충: OODA 루프는 1970년대 존 보이드가 공중전 의사결정 모델로 제시한 개념

KEY INSIGHT 기존 스킬은 그대로 필요하고 AI를 쓰며 일하는 능력이 새로 필요함 — 도입 초기에는 일시적 효율 저하를 감안해야 함



결과물 품질을 높이면서 업무 효율도 함께 개선

AI를 알맞게 활용하는 것이 전제

그대로 중요한 스킬

프로그래밍 · 노코드 · 영업 스킬

필요한 스킬셋은 크게 바뀌지 않음

새로 필요한 능력

AI를 활용하며 업무를 수행하는 능력

가장 크게 바뀌는 부분

도입 초기 주의

익숙해지기 전까지 효율이 일시 저하

초기 성과로 성급히 판단하지 않음

Source: 원 발표 10쪽(요시자키 료스케, 2024.5.25)

5.5 클라우드 때와 같은 변화가 AI로 반복됨

KEY INSIGHT AI가 닿지 않는 안전지대는 없음 — 클라우드 때처럼 AI와 함께 일하는 쪽을 택해야 역할이 넓어짐

과거: 클라우드 등장

클라우드 → 프로그래머에게 기회가 됨 → **IaC** (Infrastructure as Code)

지금: AI 도구 등장

AI 도구 → 같은 기회가 다시 옴 → **개발 전 과정** (코딩 밖 모든 장면에서 AI 활용)

AI와 경쟁하는 구도

AI가 못 하는 안전지대를 찾는 방식
AI는 코딩 밖의 일에도 쓰이므로 찾기 어려움

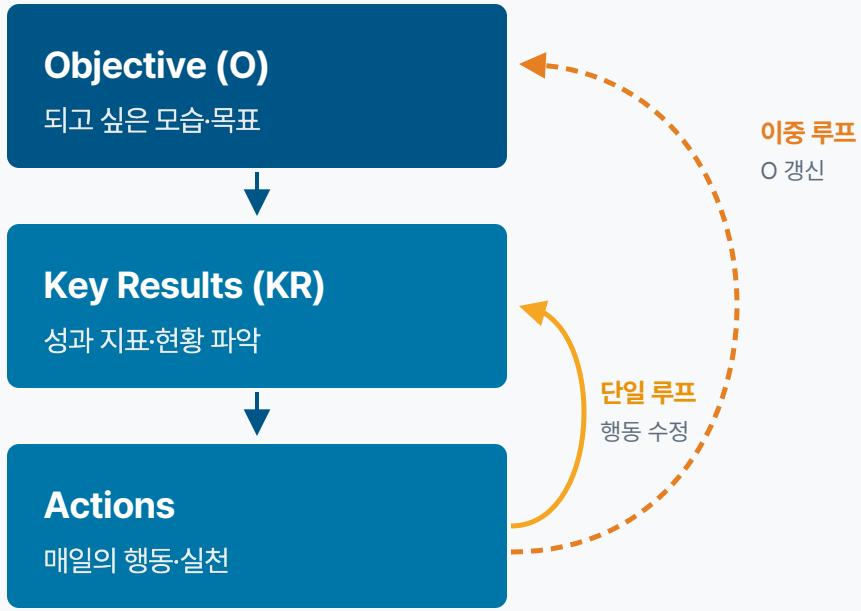
AI와 함께하는 구도

AI를 도구로 받아들여 혼자 할 수 있는
일의 폭을 넓히는 방식

Source: 구라누키 요시히토(소닉가든), Scrum Fest Sendai 2026 기조연설(2026.7.10)

KEY INSIGHT 행동을 고치는 단일 루프와 목표 자체를 고치는 이중 루프를 함께 돌려야 함 — 커리어 방향을 정기적으로 점검할 수 있음

개인 OKR 구조와 두 가지 학습 고리



이중 루프: 커리어를 정기 점검하고
목표(O) 자체를 고침

예: 자신이라는 제품의 OKR (발표 자료)

제품 목표 — 자신이라는 제품이 더 매력적이 된 모습

'설계부터 맡기고 싶다'고
의지가 되는 엔지니어

Objective — 이번 분기에 어떤 가치를 전할 것인가

팀의 설계 판단을 리드함

Key Results — 그 가치가 전해졌음을 아는 상태

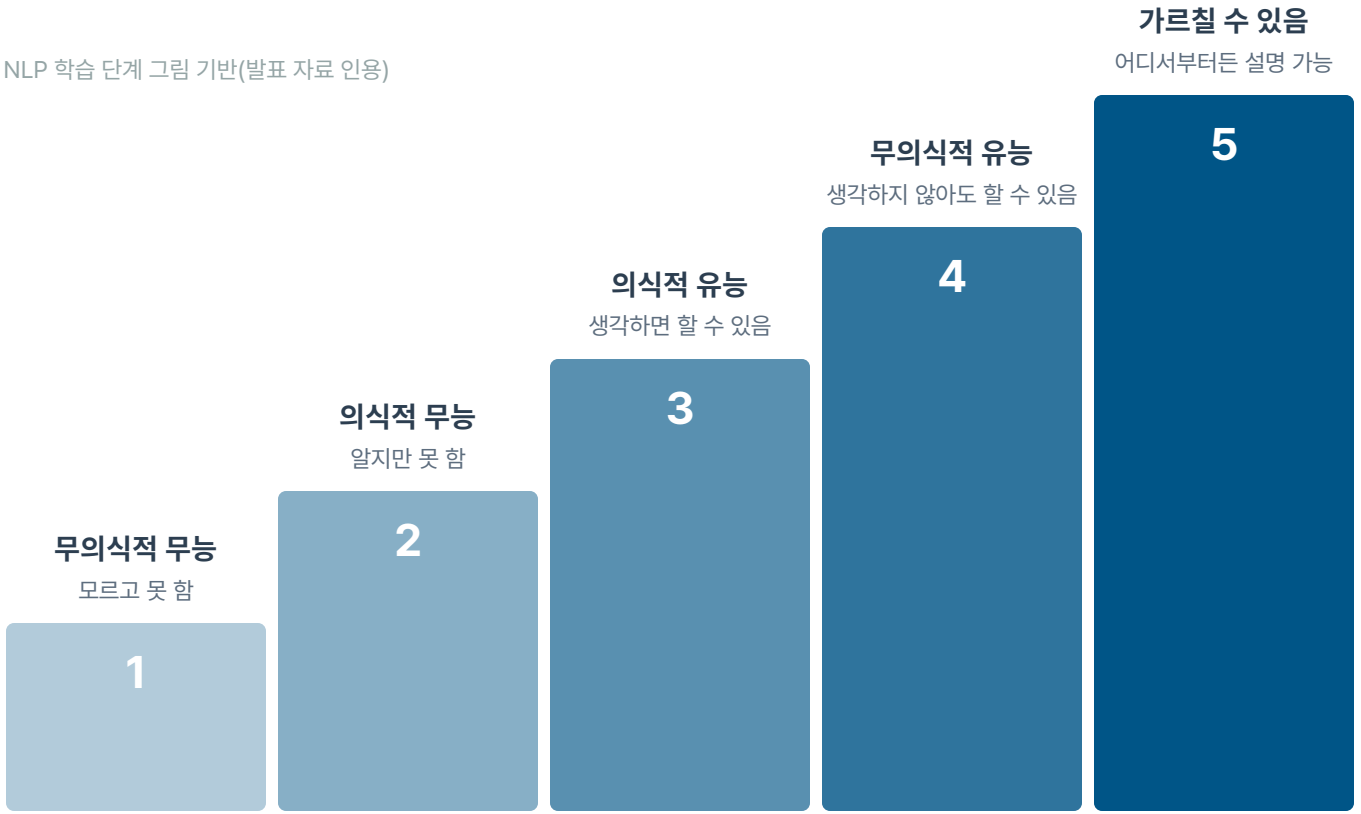
주요 설계 판단의 절반을 직접 기안
기안한 설계의 재작업 0건
설계 상담 지명 월 5건

Source: 원 발표 65~66쪽(5장 커리어 관리)

KEY INSIGHT 깊게 갈지 넓게 갈지는 정답이 없음 — 직접 고른 방향이 이후 성장 목표를 정하는 기준이 됨

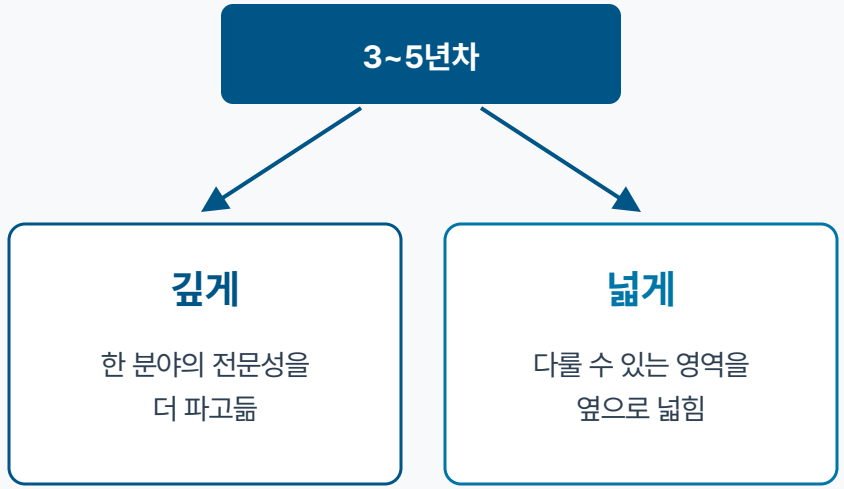
학습 5단계 — 성장은 '모르고 못 하는' 단계에서 시작

NLP 학습 단계 그림 기반(발표 자료 인용)



한 번 해낼 수 있게 된 다음은

3~5년차에 만나는 선택



어느 쪽을 골라도 됨

Source: 원 발표 49~51쪽(4장 스킬 관리, NLP 학습 단계 그림 인용)

SECTION 06

역할별 AI 사용법

맡기는 기본·위임과 검수·사무직·개발자·디자이너·기획자·조직 실행 순서

SECTION 6.1

AI에 일을 맡기는 기본

목적·전제·완료 기준을 주고 결과를 검수

6.1 AI 초안과 검수의 기본 원칙

KEY INSIGHT AI를 회사 사정을 모르는 신규 팀원으로 대함 — 목적·전제·제약을 처음에 넘겨야 결과의 질이 오름

잘 쓰는 방식 DO

- 목적·전제·제약·완성 모습을 처음에 전달
- 초안을 먼저 받고 자신은 편집자 역할
- 이 안의 허점 5개를 대라고 반론 요청
- 사실·수치는 반드시 직접 확인

막히는 방식 DON'T

- 한 줄 던지고 원하는 답이 아니면 포기
- 출력을 검수 없이 그대로 제출
- 판단 자체를 AI에 맡김
- 기밀 정보 취급 기준을 확인하지 않고 붙여 넣음

책임의 위치

내놓은 결과물의 책임은 AI에 있지 않고 내놓은 사람에게 있음 — 이 원칙만은 바뀌지 않음

Source: 원 발표 26쪽(Shengbo Xu, Xincere, Inc., 2026.08)

6.1 목적 언어화의 네 가지 요점

KEY INSIGHT AI를 잘 다루는 힘과 일을 정확히 하는 힘이 같은 습관에서 나옴 — 평소 업무 목적부터 말로 적는 연습이 필요함

1

말의 뜻은 문맥이 있어야 정해짐

낱말 하나로는 뜻이 정해지지 않음 (언어 게임)

2

목적을 정확히 전하면 AI를 원하는 방향으로 다룰 수 있음

지시와 함께 목적을 주면 답의 범위가 좁혀짐

3

목적은 의식하지 않으면 말로 정리되지 않음

평소 업무의 목적을 말로 정리하는 습관 필요

4

목적을 말로 정리하면 업무 품질도 함께 향상

AI의 답뿐 아니라 자신의 일의 정확도가 올라감

Source: MinoDriven (DMM.com), ai-context-verbalization 발표 자료(2025.10.23) 정리 쪽

네 요점은 원 발표 정리 쪽 기준

6.1 목적을 알려 주면 AI 답변 정확도가 오름

KEY INSIGHT 목적과 그 배경 상황을 함께 주면 AI가 맞는 답을 고르기 쉬워짐 — 목적 중심 설계를 지시에 담을 근거가 됨

목적을 먼저 적는 지시 형식

 # 목적

〇〇 를 하고 싶음

 # 지시

당신이 할 일은 ××

목적이 없으면 AI는 ××만 보고
그럴듯한 답 하나를 고름

잘 알려진 지시 방법

AI에 목적을 전하면 답변 정확도가 오름

발표 자료는 이것을 목적을 꺼낸 가장 중요한 이유로 꼽음

그 연장선의 컨텍스트 엔지니어링

목적 뒤에 있는 전제와 상황을 AI에 함께 전달해
더 정확한 답을 만들게 하는 접근

목적	전제	상황
무엇을 위해	지켜야 할 조건	현재 코드·업무 배경

보충: 실무 적용

· 저장소의 규칙 파일·설계 문서에 모듈별 목적을 적어 둠
· 작업 지시에 이슈의 목적·목표를 함께 붙여 넣음
· 리뷰할 때 "이 클래스의 목적은 하나인가"를 확인

AI 코딩 도구 대부분이 저장소 규칙 파일을 읽어 컨텍스트로 씀

Source: 원 발표 자료 50쪽

우측 아래 '실무 적용' 상자는 보충 내용이며 원 발표에는 없음

KEY INSIGHT AI 기법보다 논리적 사고와 설계 원칙이 결과를 가름 — 매일 쓰는 이슈부터 목적을 적어야 그 습관이 설계로 이어짐

GitHub Issue·작업 티켓 작성 예

검색 결과 화면 표시 속도 개선

 # 목적

사용자가 앱을 쾌적하게 쓰도록 해
이탈률을 낮춤

 # 목표 (목적의 달성 조건)

주요 검색어 결과 화면의 첫 표시 시간
2초 이내 달성

 # 수단

쿼리 실행 계획을 분석해
불필요한 JOIN 제거

발표 자료의 결론

AI 기법 대신 목적·목표·수단이라는
정리 방법과 캡슐화·DbC 같은
설계 원칙만 다룸

**AI 시대에도 결과를 가르는 것은
논리적 사고와 설계 원칙**

바로 적용할 수 있는 것

- 이슈 템플릿에 목적·목표·수단 칸 추가
- 목적 칸을 AI 작업 지시에 그대로 붙여 넣기
- 목적이 둘 이상인 클래스를 리뷰에서 표시

둘째·셋째 항목은 보충

Source: 원 발표 자료 64~72쪽

이슈 예시는 원 발표의 예를 한국어로 옮긴 것임. 원 발표는 저서 '좋은 코드/나쁜 코드로 배우는 설계 입문(개정 신판)'(기술평론사, 2024)을 참고서로 소개함

6.1 AI 활용 심화 팁

KEY INSIGHT 결과는 프롬프트 기교보다 넘기는 정보의 양과 질로 정해짐 — 전제 정보 파일 하나가 가장 큰 차이를 만듦

1 전제 정보를 자산으로 만들기

회사·고객·지난 회의록·자신의 글 습관을 한 파일로 모아 매번 함께 전달

가장 큰 차이

2 빈 화면 앞에서 보내는 시간 없애기

0→1은 AI, 1→10은 사람 — 착수 부담이 사라져 미루는 습관이 줄어듦

3 AI에 반대 심문 맡기기

고객은 어디서 거절할지, 이 견적의 허술한 전제가 무엇인지 묻기 — 자신의 사각지대가 보임

4 반드시 3안을 받아 직접 고르기

1안으로는 좋고 나쁨을 판단할 수 없음 — 고르는 횟수가 곧 우선순위 판단 연습

5 생긴 시간을 다시 작업으로 채우지 않기

고객 만나기·현장 보기·결정하기처럼 AI가 할 수 없는 일에 다시 투자

Source: 원 발표 27쪽(Shengbo Xu, Xincere, Inc., 2026.08)

6.1 프롬프트에 원칙만 적으면 결과가 부족함

KEY INSIGHT 빠진 개념은 목적임 — 목적이 아키텍처와 모듈 구조를 크게 좌우하기 때문

흔한 기대

변경 용이성과 아키텍처를 프롬프트에 적어 지시

"원칙을 적어 두면 AI가 알아서 지켜 줄 것"

→

실제 결과

불충분한 답이 돌아옴

원칙 이름만으로는 어디서 코드를 나눌지 정하지 못함

지속 개발에는 과제가 남음

→

빠진 개념

목적

아키텍처와 모듈 구조에 큰 영향을 주는 요소

→ 목적 중심 설계

무엇을 위한 코드인지 알려 주지 않으면 AI도 사람도 코드를 나눌 기준이 없음
원칙 이름보다 그 코드가 이루려는 목적을 먼저 정의하는 일이 필요함

Source: 원 발표 자료 17~23쪽

'불충분한 답이 돌아온다'는 원 발표자의 경험에 근거한 판단임

SECTION 6.2

위임 범위와 검수

검증 가능 범위가 정하는 위임의 상한

6.2 판단력을 키우는 실무 습관

KEY INSIGHT 작은 단위로 맡기고 매번 검증해야 판단 근거가 쌓임 — 큰 단위로 맡기면 검증 경험 자체가 사라짐

1 직접 설명할 수 있을 때까지 확인

"왜 이 구성인가"를 자기 말로 설명하지
못하면 내보내지 않음

2 작게 맡기고 매번 검증

변경은 작은 단위로 나누고 배포·테스트·
what-if로 매번 확인

3 여러 모델·관점으로 비교

다른 모델이나 동료에게 같은 질문을 던지고
답이 엇갈리는 지점을 단서로 삼음

4 분석·계획에 고성능 추론 모델 사용

설계와 방침 수립은 고성능 추론 모델과 하고
결과는 반드시 사람이 리뷰

5 장애 회고에 참여

무슨 일이 왜 일어났고 왜 놓쳤는지가
다음 판단의 재료가 됨

6 AI 결과물을 꼼꼼히 리뷰

권한·공개 범위·비용처럼 영향이 큰
곳부터 확인

Source: 원 발표 p.19

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

6.2 AI에 맡길 일과 직접 지킬 일

KEY INSIGHT 위임 범위가 넓어질수록 놓지 않을 영역을 먼저 정해야 함 — 그 경계가 스킬 공백과 동기 상실을 막음

AI에 맡길 수 있는 일이 늘어난 만큼 '내가 놓지 않을 것'을 정하는 기준 세 가지

스킬

AI에 맡길 영역과 내 것으로 만들 영역의 구분

점검 예(보충)

설계 판단·장애 분석은 직접,
반복 코드 생성은 AI에

가치관

무엇을 소중히 여기며 일하고 싶은가

점검 예(보충)

품질·사용자 경험·팀 성장 중 양보하지 않을 것 하나 고르기

동기

왜 그 일을 하는가

점검 예(보충)

AI가 대신해도 직접 하고 싶은 일이 무엇인지 적어 보기

Source: 원 발표 19쪽(무엇을 AI에 맡기고 무엇을 직접 가질까)

6.2 업무 판단을 AI와 수시로 상의하기

KEY INSIGHT AI와 상의하는 데는 비용이 거의 들지 않음 — 따를지는 본인이 판단하면 되므로 질문을 아낄 이유가 없음

상의할 질문 예시

코드

어떤 코드가 좋은가

작성한 코드의 개선점을 먼저 물어봄

설계

어떤 설계가 좋은가

대안 설계와 각각의 장단점을 비교함

메시지

어떤 문안의 메시지가 좋은가

보내기 전에 표현과 어조를 점검함

사내 조율

누구에게 먼저 이야기할까

의사결정 순서와 관계자를 미리 정리함

AI의 답을 그대로 따를 필요는 전혀 없음 — 상의 자체에 비용이 들지 않으니 적극적으로 활용

Source: 원 발표(스리셰이크, 2026.02.04) 철저하게 AI와 상의하기 쪽

질문 네 가지는 원 발표 기준. 카드 아래 설명 문구는 보충

KEY INSIGHT 같은 결과물도 시스템 관점 점수와 사람 관점 점수가 다르게 나옴 — 두 점수의 차이를 메우는 일이 지금의 엔지니어링 업무

Mark의 기준 — 시스템 관점

Mark Russinovich, Azure CTO

Correctness

부하가 걸려도 주장대로 동작하는가

Performance

p99 지연·대규모 부하에서 어떻게 동작하는가

Architecture

다음 사람이 바꿔도 구조가 유지되는가

같은 결과물

Scott의 기준 — 사람 관점

Scott Hanselman, Microsoft

Clarity

지친 사람이 읽어도 이해하고 믿을 수 있는가

Confidence

동작한다는 사실뿐 아니라 동작하는 이유까지 이해하는가

Recoverability

새벽 2시에 장애가 나도 고칠 수 있는가

같은 답을 두 개의 평가표로 채점함

발표 요지: 두 평가표 사이의 차이를 다루는 곳에 지금의 엔지니어링이 있음

Source: 원 발표 p.17 — Mark's Rubric · Scott's Rubric

원 발표: 이노우에 아키라(일본 마이크로소프트), JAZUG 16주년, 2026-10-03

KEY INSIGHT AI에게 설계까지 맡기더라도 그렇게 맡기기로 정한 것은 사람임 — 출발점과 판단 주체는 바뀌지 않음

전략

어느 작업은 AI로,
어느 작업은 사람이
할지 정함

예: 리스크 도출은 AI,
품질 목표는 사람

프로세스 설계

AI를 어떤 순서와
방식으로 적용할지
정함

예: 공정마다 산출물과
리뷰를 닫고 넘김

환경 설계

AI의 권한과
가드레일을 정함

예: 쓸 수 있는 도구와
넣으면 안 되는 정보

책임

AI의 동작과 결과에
대한 책임은 판단한
사람에게 있음

예: 테스트 완료 판정은
사람이 내림

이 넷을 모두 AI에 생각하게 하는 경우

AI에게 전부 생각하게 하는 프로세스 자체를 정한 것도 사람임. 출발점은 언제나 사람이고, 목적을 위해 도구를 고르고 씀

Source: 원 발표(MIXI, 2026.07.23) 사람이 하는 일 쪽

카드 하단의 회색 예시는 테스트 업무에 맞춘 보충

6.2 검증할 수 있는 범위까지만 위임

KEY INSIGHT 업무의 60%에 AI를 쓰는 엔지니어도 완전히 맡길 수 있는 업무는 0~20%로 봄 — 위임 대상은 검증 담당자와 판정 기준이 있는 업무부터 고름

약 60%

AI를 쓰는 업무 비중

0~20%

완전히 맡길 수 있는 업무

구분	판정 기준이 명확함	판정 기준이 모호함
검증 담당자 있음	1순위 — 바로 위임하고 담당자가 표본 검수	2순위 — 판정 기준을 문서로 정한 뒤 위임
검증 담당자 없음	3순위 — 검증 담당자를 지정한 뒤 위임	보류 — 기준부터 마련
오류 비용이 큼	한 단계 낮춰 부분 위임으로 시작	위임하지 않음

Source: Anthropic, How AI is transforming work at Anthropic(2025.12, 132명 설문·53명 인터뷰) · MSAP.ai 백서 3.1.2절

SECTION 6.3

사무직의 AI 사용법

문서·보고·데이터 업무에서 바로 쓰는 방법

6.3 사무 업무별 AI와 사람의 분담

KEY INSIGHT 초안·요약·정리는 AI에 맡기고 사실 확인·판단·발송은 사람이 맡음 — 업무마다 사람이 판정할 지점을 먼저 정해 두면 검수가 빨라짐

업무	AI에 맡길 부분	사람이 판정할 부분
문서·보고서	목차 초안, 자료 요약, 문장 다듬기	결론과 수치가 원자료와 맞는지, 보고 대상에 맞는지
메일·안내문	초안 작성, 어조 조정, 번역	수신자·첨부·약속 내용 확인 뒤 발송
회의	녹취 정리, 결정 사항·할 일 추출	결정 사항의 표현과 담당자 지정
데이터 정리	표 변환, 분류, 중복 찾기, 수식 작성	기준 정의와 결과 표본 검사
조사	자료 수집과 비교표 초안	출처 확인과 결론 채택

Copilot 현장 실험(66개 기업 7,137명): 이메일에 쓰는 시간이 주당 약 3시간 줄었고 회의 시간은 그대로였음

Source: MSAP.ai 백서 1.2·3.1절 · Dillon 외, Shifting Work Patterns with Generative AI, NBER WP 33795(2025, Copilot 7,137명 실험)

KEY INSIGHT 직원 조작은 붙여 넣기 한 번으로 끝나고 AI가 질문하며 활용안까지 이끌어 냄 — 편집이나 고민을 요구하면 첫 시도에서 이탈함

AI 체험 가이드: 복사해 붙이는 프롬프트

직원은 복사해서 붙여 넣기만 함
AI가 질문하며 업무 상황을 끌어내고
구체적 활용안과 결과물까지 이끌어 냄

운영 원칙

- 편집시키지 않음
- 고민시키지 않음
- 복사 한 번으로 시작

먼저 메일 초안 · 요약 · FAQ 초안 같은 편의 프롬프트 사용
AI의 편리함을 체감한 뒤 체험 가이드 대화 시작

대화 흐름 (AI가 묻는 순서)

1. 업무 내용 확인
2. 반복 작업 파악
3. 결과물 · 관계자 파악
4. 불편 · 제약 정리
5. 활용안 제시

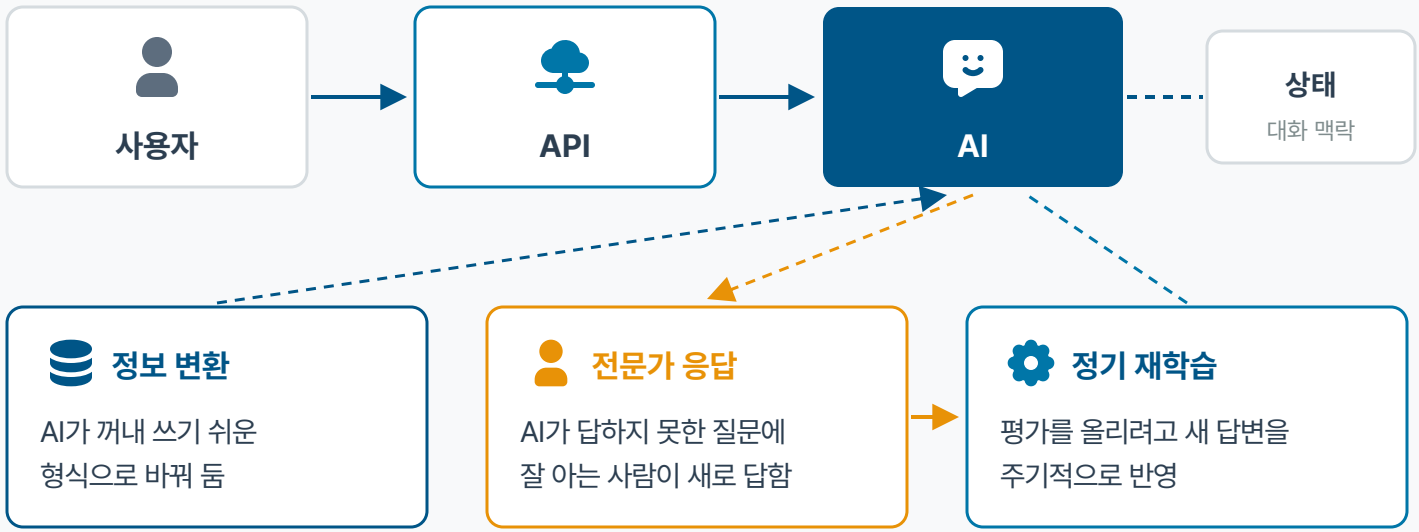
발표 자료 사례: 파나소닉 커넥트

일상 업무 15건의 프롬프트 예시를 첫 화면에 두고 바로 쓰게 함 (발표 자료 기준, 세부 수치 미확인)

Source: 일본 오라클 OCI Platform COE 본부 솔루션즈 아키텍트, AI 주도 AI 확산 활동(AAAP) 발표 자료, 2026 7쪽

KEY INSIGHT AI 단독으로는 업무 환경을 모름 — 정보 변환·전문가 응답·정기 재학습을 함께 갖춰야 답변 품질이 올라감

업무와 AI 사이에 필요한 구성



AI가 많은 부분을 블랙박스로 처리해 주지만 업무에 투입하려면 육성이 필요함

사람처럼 실패도 있고, 처음에는 놓인 환경을 알려 주어 차이를 메워야 함

발표 요지

AI는 아직 시스템의 일부임

성능 향상과 이용 비용 하락에도
업무 투입 전 단계의
간극은 여전히 남아 있음

사람과 같은 육성 과정

- 환경 설명 → 정보 변환
- 선임자의 답 → 전문가 응답
- 경험 축적 → 정기 재학습

대응 관계는 이해를 돕는 요약이며
발표 자료 도식을 다시 그린 것임

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 27쪽

KEY INSIGHT 구어로 풀어낸 맥락을 추론형 모델이 작업 절차로 정리함 — 프롬프트를 정교하게 짜는 부담이 줄어듦

말로 요청하기

말로 설명하면 배경·제약·의도가
글로 쓸 때보다 풍부하게 들어감

말로 요청할수록 정확도가 오른다는
LLM의 특성과 잘 맞음(발표 자료 주장)

+

Thinking 모델

Claude 3.7 Sonnet 같은 추론형 모델을
함께 쓰면 작업 절차까지 제안

프롬프트를 단계별로 세세하게
짜 두지 않아도 됨

결과

암묵지를 형식지로 바꾸는 작업을 더 적은 준비로, 더 효율적으로 처리

Source: 원 발표 자료(원커리어, 2025.04.23) 17쪽 요지

'말로 요청하면 정확도가 오른다'는 원 발표의 주장이며 외부 근거는 확인하지 못함(원 발표 기준, 미확인)

KEY INSIGHT AI 출력을 JSON으로 고정하면 기존 애플리케이션과 워크플로에 바로 연결됨 — 에이전트 전에 예측 가능한 범위에서 운영 경험이 쌓임

예시: 발표 자료 작성 자동화

준비 단계 (사람)

- **PowerPoint** 템플릿 준비
- **슬라이드** 사양서 준비

손으로 하는 꾸준한 작업

사용 단계

발표 원고 (입력) → 레이아웃 선택 (AI) → 형식 맞춘 JSON (AI) → 자리표시자 채움 (python-pptx) → 손질 후 완성 (사람)

```
{
  "title": "AI와 협업하는 인재 ...",
  "subtitle": "AI의 성질을 알기 ...",
  "items": [ { "desc": "..." }, ... ]
}
```

데이터 형식으로 주고받으면
기존 앱과 워크플로에
그대로 넣을 수 있음

순서

AI 에이전트 기반 워크플로로 발전하겠지만
먼저 예측 가능한 범위에서 AI를 다루는 데 익숙해짐

도구 선택

마지막 손질은 남으므로 평소 쓰는 도구로 편집을 모음
Illustrator·Figma보다 많은 사람이 쓰는 PowerPoint가 유리

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 11쪽

6.3 아낀 시간을 처리량으로 잇는 방법

KEY INSIGHT 국내 근로자의 업무용 AI 사용은 51.8%지만 절감 시간과 처리량의 상관은 0 — 아낀 시간이 업무 절차 재설계로 이어지지 않았기 때문

51.8%

업무에 생성형 AI를 쓰는 근로자

미국의 약 2배

−3.8%

사용자의 업무시간

주 40시간 기준 약 1.5시간

상관 0

절감 시간과 처리량 증가

기업 단위 활용률은 9.6%

1 반복 업무 고르기

주마다 반복되는 업무 한 가지와 그 완료 기준을 적음

2 절차 다시 쓰기

AI가 맡을 단계와 사람이 판정할 단계를 나눠 팀 절차서에 반영

3 아낀 시간 배정

줄어든 시간을 고객 대응·분석 같은 업무에 공식적으로 배정하고 결과를 기록

Source: 한국은행, AI의 빠른 확산과 생산성 효과(BOK 이슈노트, 2025.08) · 한국은행(서동현), AI 도입은 생산성을 높이는가?(BOK 이슈노트 제2026-12호, 2026.06, 집필자 견해)

SECTION 6.4

개발자의 AI 사용법

명세·규칙 파일·작은 작업 단위

KEY INSIGHT 탐색 → 계획 → 구현 → 검증 순서로 맡기고 에이전트가 직접 돌려 볼 검사를 줌 — 꼭 지킬 규칙은 부탁하지 말고 권한으로 막음

1 탐색

관련 코드와 문서를 읽게 하고 아직 고치지 않게 함

2 계획

변경 범위와 순서를 적게 하고 사람이 확인

3 구현

작은 단위로 나눠 맡기고 금지 파일·배포 절차를 규칙 파일로 알림

4 검증

테스트·명세 대조를 에이전트가 직접 돌리게 하고, 꼭 지킬 규칙은 권한으로 막음

리뷰·테스트 시간을 작업 계획에 처음부터 넣음 — 거의 맞지만 조금 틀린 결과를 걸러 내는 단계가 품질을 정함

Source: Anthropic, Best practices for Claude Code(2025~2026)

6.4 Web API 개발 단계별 사람의 확인 지점

KEY INSIGHT 사양·설계·스키마에서 입출력 범위를 좁혀 두면 구현과 테스트를 AI에 맡길 수 있음 — 사람의 확인 범위가 줄어 AI 속도가 살아남

사람이 집중할 영역 — 입출력 범위를 명확히 정함 | **AI에 맡기는 영역**

사양서 작성	처리 설계	스키마 설계	처리 구현	테스트 설계·구현
OpenAPI 형식 엔드포인트와 입출력을 명확히 적음 상태 코드까지 명시	**순서도** mermaid.js 또는 eraser.io로 그림 눈으로 확인해 인지 부담을 줄임	**스키마 정의·검증** 함수 입출력 타입과 값 범위를 좁힘 Next.js는 Zod Python은 Pydantic	**AI가 구현** 앞 단계의 정보를 받아 코드를 작성 테스트 요건 충족을 자동으로 확인	**AI가 테스트 작성** 테스트 스위트 → 테스트 케이스 → 테스트 코드 자동 검증

발표 요지

예전에는 컴퓨터에 지시하려고 사람이 프로그램을 썼지만 지금은 AI가 씀
사람은 AI 속도를 막지 않도록 무엇을 어디까지 확인하면 품질이 보장되는지 정함

Source: 요시자키 료스케(Wadan Inc.), 'AI 발전에 따른 패러다임 전환' 발표 자료(2024-06-14) 9쪽

KEY INSIGHT 도구마다 다르던 규칙 파일이 한 형식으로 모임 — LayerX처럼 문서를 리포지토리에 둔 팀은 바로 옮겨 쓸 수 있음

repo/AGENTS.md

```
# AGENTS.md

## 빌드 · 테스트
- pnpm install · pnpm test

## 코딩 규칙
- 설계 문서: docs/design-docs/ 참조
- 동작 변경과 리팩터링은 별도 PR

## 금지 사항
- migrations/ 직접 수정 금지
- 비밀 값 커밋 금지

# 예시 — 형식은 일반 Markdown
```

2025-08

OpenAI 공개 · Markdown 형식

에이전트가 읽는 README

6만 개 이상

오픈소스 프로젝트 · 프레임워크 채택

2025-12

Agentic AI Foundation(Linux Foundation)

창립 프로젝트로 편입 — MCP · goose와 함께

특정 회사 형식이 아닌 중립 규격

Source: Linux Foundation 보도자료(2025-12-09), agents.md

[보강] 원본 이후 공개 자료로 추가한 장

KEY INSIGHT 계획 먼저, 동작 변경과 구조 변경 분리, 테스트 먼저는 모두 한 번에 처리할 양을 줄이는 방법임 — 요령이 아니라 인지 부하 관리로 설명됨

스스로 던질 점검 질문

Q1 AI가 모르는 암묵 사양은 없음?

사람 머릿속에만 있는 규칙 확인

Q2 데이터 스키마를 넘겼음?

테이블 · 타입 정의 제공

Q3 기존 동작을 깨지 않았음?

회귀 테스트 · 변경 범위 확인

AI 코딩에서 공유되는 관행

계획부터 세우게 함

작업을 덩어리로 나눠 한 번에 볼 양을 줄임

동작 변경과 구조 변경 분리

리팩터링과 기능 변경을 섞지 않음

테스트부터 시작

가드레일을 먼저 세우고, 검증 결과는 사람이 확인

Source: 나카가와 요시키(LayerX) 발표(2025-07) 원본 p.33, AI Coding Meetup #2

6.4 숙련 단계별 AI 활용 사례

KEY INSIGHT 초급은 업무 정비, 중급은 인접 영역, 상급은 도구 연결임 — 단계마다 손댈 곳이 달라 팀의 현재 위치부터 정해야 함

초급 · 지금 업무 정비

리스크 기반 테스트 계획
사양에서 리스크를 뽑고 접근법을 정함

추적성 확립
사양·리스크·관점·설계·항목을 연결

써 보지 못한 기법 시도
테스트 매트릭스, VSTeP 등

하고 싶던 일을 AI로 해 봄

중급 · 인접 영역 진출

API 테스트
UI 뒤의 API 응답으로 결과 확인

개발자 테스트 확인
유스케이스별 커버리지와 빈틈을 질의

테스트 보조 도구 개발
준비 작업·결과 정리 도구를 직접 만듦

기존 영역에서 한 걸음 밖으로 나감

상급 · 흩어진 도구 연결

CI/CD 형 테스트 파이프라인
관점 생성·케이스 생성·E2E를 자동 연결

제품 전체 테스트 계획
로직은 Unit test, 데이터는 API·리뷰,
UI는 E2E로 레벨 전체를 나눠 계획

개발 프로세스 전체로 품질 활동 확대

발표 자료 사례: 터미널을 써 본 적 없던 QA 담당자

AI로 불편한 테스트 작업을 줄이는 도구를 만들고, 도구를 이어 프로세스를 구축한 뒤 개발자 코드를 읽고 질문하는 데까지 감

Source: 원 발표(MIXI, 2026.07.23) 초급·중급·상급 사례 쪽과 N 씨 사례 쪽

세 단계의 이름과 사례는 원 발표의 구분을 따름

KEY INSIGHT 에이전트 결과가 늘 맞지는 않아 사람이 검토하는 단계의 품질이 전체 품질을 정함 — 요구사항 양식·반복 검증 자동화·보안 검토 체계를 함께 갖춤

정비 1

요구사항 양식

- 목적·완료 조건·제외 범위를 적는 칸이 있는가
- 명세를 쓴 사람이 기록되는가

정비 2

반복 검증 자동화

- 에이전트가 스스로 돌릴 테스트가 있는가
- 회귀 확인이 사람 손 없이 도는가

정비 3

보안·품질 검토

- 보안 검토가 필요한 변경은 사람 승인 없이 배포되지 않는가
- 승인한 사람이 변경마다 남는가

DORA 2025: AI는 팀의 기존 상태를 증폭함 — 플랫폼·작은 단위 배포 같은 기반이 약하면 처리량과 함께 불안정성도 늘어남

Source: MSAP.ai, AI 시대 일하는 방식 백서 6.4.1절(2026.10) · DORA, State of AI-assisted Software Development 2025

SECTION 6.5

디자이너·기획자의 AI 사용법

디자인 데이터 정비와 에이전트 기능 기획 원칙

KEY INSIGHT 디자이너가 리뷰를 기다리지 않고 스스로 데이터를 점검·수정함 — 엔지니어 리뷰 대기가 줄고 코드 생성 결과가 안정됨



리뷰 대기 없는 자체 점검

디자이너가 엔지니어 리뷰를 기다리지 않고
스스로 품질을 점검하고 고치는 환경을 구축

발표 자료 5쪽 요지

코드 생성 입력 품질의 일정화

같은 규칙을 통과한 데이터만 코드로 바뀜.
그 결과 구현자별 품질 편차가 줄어듦

보충 해설

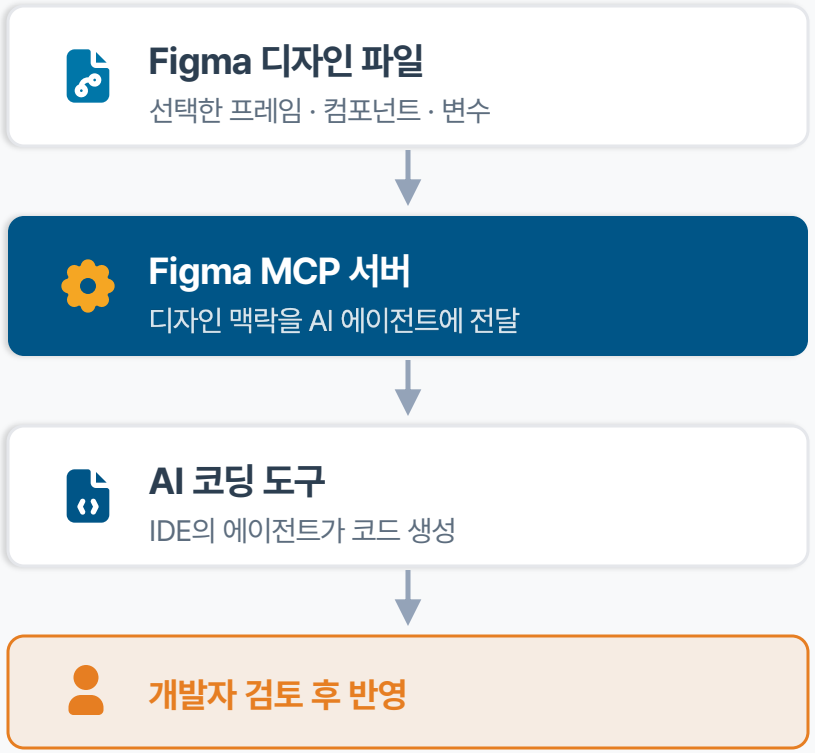
Source: Shuzo Takahashi(라인야후), 'AI와 함께 걸어가는 디자이너의 역할 변화' 5쪽

수정 되돌림 화살표와 오른쪽 효과 카드는 원 발표 흐름도를 풀어 쓴 보충 설명

Figma MCP 서버의 디자인 데이터 전달

KEY INSIGHT 공개 도구로도 디자인 데이터를 AI 코드 생성에 바로 넘길 수 있음 — 다만 결과 코드는 개발자 검토를 거쳐야 함

구성



공식 가이드의 주요 기능

선택 프레임 기반 코드 생성

변수 · 컴포넌트 · 레이아웃 정보를 IDE로 가져와 코드 작성에 사용

create_design_system_rules

프런트엔드 코드 생성에 쓸 디자인 시스템 규칙 파일을 만듦

Code Connect

Figma 컴포넌트를 저장소의 실제 코드 컴포넌트와 연결

라인야후의 Figma Linter처럼 입력 데이터 규칙이 먼저 정리돼 있어야 결과가 안정됨

Source: github.com/figma/mcp-server-guide, help.figma.com 'Guide to the Figma MCP server' (2026-10 확인)

보충: 원 발표에 없는 공개 도구 정리. 공식 가이드는 일부 기능을 베타로 표기하며 기능 범위는 바뀔 수 있음

6.5 디자이너의 AI 사용 현황과 사용 범위

KEY INSIGHT 효율이 오른다는 응답은 78%지만 결과를 믿고 쓴다는 응답은 32% — 리서치 요약·변형안처럼 범위가 좁은 작업부터 맡기고 화면 판단은 디자이너가 함

78%

AI로 효율이 오름

32%

AI 결과를 믿고 씀

69% vs 82%

AI 도구 만족도

디자이너 vs 개발자

구분	내용
AI에 맡길 일	리서치 요약, 문구·레이아웃 변형안, 동작하는 시제품 초안, 디자인 규칙 점검
디자이너가 지킬 일	정보 구조·접근성 판단, 브랜드에 맞는지 결정, 화면 전체 결과의 수정

Source: Figma, 2025 AI Report(2025) · Nielsen Norman Group, State of UX 2025

KEY INSIGHT 어려운 판단을 한 번에 맡기면 출력이 흔들림 — 예·아니오 수준까지 나눠야 결과를 검증하고 안정시킬 수 있음

대책 1

난이도 낮음 — 난이도 높음

2값 분류

다음 일정에 고객이 포함되는지 판단:
LayerX 상담 / 참석자 AAA

예 또는 아니오

답이 좁아 맞고 틀림을 바로 확인 가능

나눠서 맡김

복잡한 판단·생성

내일 전략 회의에 부장이 못 오게 되었으니
참석자 일정을 확인해 가능한 후보 목록을
이유와 함께 제시

응답 형태: 일정 확인 결과와 후보별 이유를 담은 긴 글

여러 판단이 한 응답에 섞여 결과가 매번 달라짐

발표 요지: 아주 어려운 일을 한 번에 시키고 있을 가능성이 있음, 한 번에 맡기는 문제를 작게 해야 출력이 안정됨

Source: Kenta Watanabe (LayerX), pmconf 2025 발표 자료 (2025.12.04)

두 예시는 원 발표 기준. 카드 아래 회색 설명은 보충

KEY INSIGHT 사용자가 무엇을 시킬지 고민하지 않아도 됨 — 실행 전에 사용자가 확인하므로 오류를 걸러 낼 수 있음

일정 최적화 제안 (예시)

팀 정기 회의가 주 3시간 이내로 유지됨

개발팀 정기 회의는 전사 평균 2.5시간

상태 알림

다음 주 월·화 회의 사이 자투리 시간 정리

30분 이상 쪼개진 시간이 4곳 있음

개선 제안

파트너 정기 회의 일정 변경이 필요할 수 있음

팀원 Taro 님이 휴가에 들어감

Taro 님 외 사내 참석자가 없는데 일정은 그대로임

위험 알림

사람이 하는 일

승인

수정

수정 내용 입력

채팅 방식과의 차이

채팅: 사용자가 먼저 요청을 떠올려 입력

제안: 에이전트가 일정을 분석해 먼저 보여 줌

사람은 결과를 보고 고르기만 함

변경은 승인한 뒤에만 실행

보충: 승인 단계가 HITL 역할을 함

Source: Kenta Watanabe (LayerX), pmconf 2025 발표 자료 (2025.12.04)

제안 세 건과 버튼은 원 발표 화면 예시를 도식으로 다시 그림. 수치는 원 발표 예시값. 오른쪽 아래 비교 설명은 보충

KEY INSIGHT 세 가지 모두 불확실성을 기획 단계에서 먼저 다루는 일임 — 모델 성능 향상을 기다리지 않고 지금 기술로 많은 업무를 바꿀 수 있음

01

LLM 동작을 경험으로 이해

LLM 동작의 해상도를 높임

문제 분해, 자사 문제 평가,
제품용 프롬프트 직접 운영

02

LLM보다 먼저 사람을 이해

사람의 생각을 따라가며 깊이 조사

암묵적 규칙 발굴, 판단 기준을
컨텍스트로 제공

03

LLM에 풀게 하지 않는 판단

LLM에 맡기는 일을 줄이는 결정

코드로 옮기기, 입출력 화면 설계,
먼저 제안하기, 작업 범위 좁히기

발표 자료 맺음말 요지: 범용 인공지능 논의와 별개로 지금 있는 기술로 많은 부분이 바뀜
벽은 많지만 경험을 빠르게 쌓는 것 외에 길이 없음, 컨텍스트 엔지니어링·평가(Evals)도 같은 과제

Source: Kenta Watanabe (LayerX), pmconf 2025 발표 자료 (2025.12.04)

세 원칙과 한 줄 설명은 원 발표 정리 쪽 기준. 카드 아래 회색 상자는 앞 장 내용을 묶은 보충

SECTION 6.6

조직의 AI 도입 실행 순서

한 부서 한 업무로 시작하는 4단계

6.6 AX 4단계 실행 순서와 단계별 담당

KEY INSIGHT 목적 정의 → 업무 전문가의 에이전트 제작 → AI Enablement Lead 지정 → 손익 기준 지표 — 결과 단위는 1단계 문서에 미리 적어 둠

단계	주 담당	이 단계에서 결정할 것	첫 분기 산출물
1. 목적 정의	부서장(경영진 승인)	위임할 업무, 완료 기준, 검증 담당자	부서별 위임 업무 목록
2. 업무 전문가의 에이전트 제작	실무자(Agent boss)	누가 어떤 에이전트를 만들고 검수하는가	운영 중인 에이전트와 검수 기록
3. AI Enablement Lead 지정	경영진	소속, 성과 기준, 지원 범위	확산 계획과 부서 지원 기록
4. 손익 기준 결과 지표	경영진·재무	결과 단위와 단가, 보고 주기	분기 경영 보고 항목

순서를 바꾸면 되돌리는 비용이 커짐 — 목적 없이 에이전트부터 만들면 손익 변화를 보고할 근거가 남지 않음

Source: MSAP.ai, AI 시대 일하는 방식 백서 8.1절(2026.10) · McKinsey, The State of AI 2025(2025.11)

6.6 한 부서 한 업무 시범 착수 범위

KEY INSIGHT 검증 담당자가 있고 결과 단위를 셀 수 있는 업무 하나를 한 분기 안에 끝까지 재설계함 — 정착을 확인한 뒤 다른 부서로 넓히는 순서가 위험이 가장 작음

항목	시범 착수 범위
대상	검증 담당자가 있고 결과 단위를 셀 수 있는 업무 1건
참여자	업무 전문가 1명(Agent boss), 검증 담당자, AI Enablement Lead 후보
첫 분기 지표	결과 단위 처리 건수, 처리 시간, 검수 불합격률
분기 말 판단	다른 부서로 넓힐지, 위임 범위를 조정할지 결정

대상 업무 확인 세 가지

시작 신호·판단 단계·결과 저장 위치

- 새 파일·정해진 시각·접수된 요청처럼 시작 신호가 분명한가
- 에이전트 처리 뒤 사람이 판단하는 단계가 정해져 있는가
- 결과가 위키·CRM·공유 드라이브처럼 이미 쓰는 시스템에 쌓이는가
- 예: 명함 정보 등록, 정기 보고 자료 취합

Source: MSAP.ai, AI 시대 일하는 방식 백서 8.4.2절(2026.10)

6.6 단계별 성과 지표 설계

KEY INSIGHT 채택률을 먼저 보고 업무 효과 수치는 조직 준비가 된 뒤에 확인함 — 처음부터 수치 목표를 걸면 형식적 참여만 늘어남

1단계 채택 확인

보는 지표

- AI 체험 가이드 실행자 수
- 발견 공유 노트 공유율
- AI 도구 주간 활성 사용자 수

수치 목표로 걸면 형식적 참여 위험

2단계 행동 변화 확인

보는 지표

- AI 단계가 들어간 업무 흐름 비율
- 반복 사용 빈도
- 습관이 된 작업 유무

회차 안에서 정성적으로 확인

3단계 업무 효과 확인(선택)

보는 지표

- 업무 사이클 타임 단축
- 결과물 품질 향상
- eNPS (직원 몰입도)

조직 준비가 된 시점에 검토

Source: 일본 오라클 OCI Platform COE 본부 솔루션즈 아키텍트, AI 주도 AI 확산 활동(AAAP) 발표 자료, 2026 15쪽

6.6 AI 확산 운영의 성공 조건

KEY INSIGHT 시작 절차가 복잡하면 바로 이탈이 생김 — 복사 한 번으로 시작하고 위험이 낮은 업무부터 맡김

1 시작 절차 단순화

복사 한 번으로 시작

문서를 읽고 편집하라는 순간 이탈

2 공유 채널 마련

Slack · Teams · Discord 전용 채널

발표 자료: McKinsey Lilli Clubs 역할(미확인)

3 슈퍼 유저 지정

얼리어답터를 골라

첫 성공 사례를 먼저 만듦

4 챔피언 방식의 추진

앰배서더 방식 권장

동료의 말이 지시보다 설득력이 큼

5 헬프데스크화 방지

조작 질문은 먼저 AI에게 묻도록 안내

스스로 해결하는 경로를 만듦

6 위험이 낮은 업무로 시작

요약 · 메일 초안 · FAQ 초안

실패해도 영향이 작은 업무

Source: 일본 오라클 OCI Platform COE 본부 솔루션즈 아키텍트, AI 주도 AI 확산 활동(AAAP) 발표 자료, 2026 13쪽. Lilli Clubs는 원 발표 기준이며 공개 자료로 확인하지 못함

6.6 국내 개발 팀 적용 점검 항목

KEY INSIGHT 개인 습관만으로는 늘어나는 요청을 막기 어려움 — 팀이 함께 정한 운영 규칙이 있어야 효과가 유지됨

점검 항목	팀 단위 운영 예	확인 시점
AI로 줄어든 시간의 사용처	줄어든 시간 중 학습·개선에 쓸 몫을 팀이 미리 합의	분기 계획
팀 보드의 WIP 제한	진행 중 칸에 사람당 한두 건 상한, 넘으면 새 일 착수 보류	매일 스탠드업
주간 학습 시간과 학습 채널	주 30분 공동 학습 시간과 Slack 학습 채널 운영	주간
AI 생성 코드의 원리 설명	코드 리뷰에서 작성자가 동작 원리를 한 줄로 설명	매 코드 리뷰
불안·과부하 신호 공유	회고에서 조급함·과부하를 말해도 되는 순서를 고정	스프린트 회고

운영 예와 확인 시점은 보충 예시 — 팀 상황에 맞춰 조정

Source: 보충(장 전체): 원 발표의 시간·스트레스·스킬 관리 요지를 국내 개발 팀 운영에 맞춰 정리

THANK YOU

감사합니다

AI 시대 어떻게 일할 것인가

MSAP.ai

msap.ai

https://www.msap.ai/

02-6953-5427

hello@msap.ai

© MSAP.ai. All rights reserved.