

AI를 도입한 지 1~2년 지난 조직이 지금 답해야 할 질문 네 가지를, 우리 회사 숫자로 채워서 답으로 바꾸는 진단 시트입니다. 2~4쪽을 채운 뒤 이 1쪽으로 돌아와 요약과 요청을 정리하세요. 5쪽은 계산 근거, 6~7쪽은 다른 조직의 참고 사례입니다.

작성자		작성일	
대상 조직		기준 기간	

★ 경영진 요약

2~4쪽에서 나온 숫자를 옮겨 적는다. 계산식과 판정 기준의 근거는 5쪽 부록에 있다.

지표	값	판정
미귀속 비율 (%)		
1인당 실효 부담 (원/월)		
채용 대비 손익분기 인원 (명)		
병목 신호 (2개 이상 겹침 항목 수)		

그래서 무엇을 요청하는가

예: 다음 분기 목표를 지출 절감이 아니라 귀속률 X% 달성으로 잡는다

이 문서를 읽는 법 — 1쪽은 결재 판단용 요약이다. 2~4쪽은 진단 실무자가 회사 숫자로 채우는 워크시트, 5쪽은 계산 근거와 판정 기준을 확인하는 부록, 6~7쪽은 다른 조직이 같은 문제를 어떻게 풀었는지 읽어보는 참고 자료다.

1 인벤토리 — 우리 회사는 AI에 실제로 얼마를 쓰고 있나

법인카드 · 개인 API 키 상환 · 부서별 구독을 한 표에 모은다. 셋 중 하나라도 빠지면 아래 모든 계산이 과소평가된다.

항목명	결제 방식	부서	월 단가 (원)	사용 인원	월 합계 (원)
예: ChatGPT Team	법인카드	전사			
예: Cursor	개인 키 상환	개발			
합계					

계산식 · 판정 기준 · 가정과 한계 → 5쪽 부록 A

2 귀속 — 그 돈은 누구 업무에 붙는가

1번 항목을 "누구 업무인지 특정할 수 있는가"로 다시 나눈다. 누가 썼는지만 알고 뭘 했는지 모르면 귀속이 아니다.

항목명 (1번 표에서 그대로)	월 합계 (원)	귀속 가능 여부	귀속 대상 (부서/업무)
		귀속 / 미귀속	
		귀속 / 미귀속	
		귀속 / 미귀속	
		귀속 / 미귀속	
		귀속 / 미귀속	
합계			

계산식 · 판정 기준 → 5쪽 부록 B

3 손익분기 — 사람 하나 더 뽑는 게 나은가

1번의 연 합계를, 채용 1인 연간 완전부담 비용과 같은 분모에 놓고 비교한다.

항목	값 (원)
전사 AI 연 합계 (1번 결과)	
채용 후보 직군의 연봉	
4대보험·복리후생 로딩 비율 (% , 모르면 30%)	
채용 1인 연간 완전부담 비용	
채용 대비 손익분기 인원 (명)	

계산식 · 판정 기준 · 가정과 한계 → 5쪽 부록 C

4 집중도 — 한 사람에게 몰려 있지 않은가

사내 AI 자동화·에이전트 워크플로를 나열하고, 만든 사람과 지금 쓰는 사람이 같은지 확인한다. 만든 사람이 유일한 운영자면 그건 자산이 아니라 병목이다.

자동화/워크플로 이름	제작자	현재 실사용자 수	제작자 부재 시 대체 운영 가능자	운영 방법 문서화 여부
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음
			있음 / 없음	있음 / 없음

병목 신호 3종 판정 기준 · 가정과 한계 → 5쪽 부록 D

✓ 병목 신호 집계

신호	개수	전체 대비 비율
신호 A — 단독 운영		
신호 B — 대체 불가		
신호 C — 미문서화		
2개 이상 겹치는 자동화 수		

A 인벤토리 (섹션 1)

계산식

- 항목별 월 합계 = 월 단가 × 사용 인원
- 전사 월 합계 = 모든 행의 월 합계 합 =SUM(F2:F5)
- 전사 연 합계 = 전사 월 합계 × 12 =합계셀*12
- 1인당 실패 부담 = 전사 월 합계 ÷ 전체 임직원 수(구독자 수 아님)
=합계셀/전체인원

판정 기준 · 가정과 한계

- 개인 키 상환 항목이 3줄 이상이면 인벤토리가 아직 불완전할 가능성이 높다.
- 직원의 81%가 회사가 승인하지 않은 AI 도구를 쓴다는 조사가 있다(UpGuard, 보안 리더 500명 · 직원 1,000명). 이 표 합계가 회사가 파악한 "AI 예산"보다 낮다면, 지출이 적은 게 아니라 안 보이는 것이다.
- 구독형 지출만 잡는다. 종량제(토론 과금)는 청구서 합계를 그대로 쓴다.
- 월 단가는 정가 기준. 연간 약정 할인이 있으면 연 금액÷12로 환산한다.

B 귀속 (섹션 2)

계산식

- 귀속 금액 합계 = "귀속" 행의 월 합계 합
- 미귀속 금액 합계 = "미귀속" 행의 월 합계 합
- 미귀속 비율(%) = 미귀속 합계 ÷ 전사 월 합계(1번) × 100
=SUMIF(귀속열, "미귀속", 합계열)/전사합계셀*100

판정 기준 · 가정과 한계

- 50% 이상: 절반 넘는 지출의 주인이 없다. 부서별 배분·ROI·남용 탐지가 원리적으로 불가능하다.
- 20~50%: 무시할 크기가 아니다. 항목을 목록화하고 다음 분기 안에 줄인다.
- 20% 미만: 대부분 설명 가능한 상태. 유지 자체가 관리 비용이라는 점은 남는다.
- "귀속 가능"의 판단 기준은 조직마다 다르다. 다른 조직과 미귀속 비율을 직접 비교하지 않는다.

C 손익분기 (섹션 3)

계산식

- 채용 1인 연간 완전부담 비용 = 연봉 × (1 + 로딩 비율)
- 채용 대비 손익분기 인원 = 전사 AI 연 합계 ÷ 채용 1인 연간 완전부담 비용
=AI연합계셀/(연봉셀*(1+로딩비율셀))

판정 기준 · 가정과 한계

- 1명 미만: AI 지출 전체가 신입 한 명 인건비보다 작다.
- 1~3명: 지출 규모가 채용과 같은 자릿수 — 대체 가능성이 아니라 규모 비교다.
- 3명 이상: 2번의 미귀속 비율도 높다면 우선순위는 채용 비교가 아니라 귀속이다.
- 로딩 비율 30%는 편의상 기본값. 실제 인사팀 기준값이 있으면 교체한다.

D 집중도 (섹션 4)

병목 신호 3종 판정

- 신호 A — 단독 운영: 실사용자 수 1명(제작자 본인)뿐인 자동화 개수
- 신호 B — 대체 불가: "대체 운영 가능자 없음" 표시 개수
- 신호 C — 미문서화: "운영 방법 문서화" 없음 표시 개수
- 엑셀: 각 조건 옆에 =COUNTIF(범위, "조건")

판정 기준 · 가정과 한계

- 세 신호 중 2개 이상 겹치는 자동화가 하나라도 있으면, 그 사람이 퇴사·휴직하는 순간 멈춘다 — 자산이 아니라 리스크다.
- 신호 A 개수가 전체의 절반을 넘으면 조직의 AI 활용이 소수 인원의 개인기에 의존하고 있다는 뜻이다.
- "실사용자 수"는 로그인 횟수가 아니라 실제 업무 산출물에 결과가 들어가는 사람 수다.

1 토스 — 확산 담당자를 조직적으로 지정한다

- 무엇을 했나: 조직마다 AI를 잘 퍼뜨리는 사람을 추천받아 AI Surf Evangelist 142명을 선발했다. 팀원들이 자율적으로 만든 사내 클럽 약 200개가 생겼고, AI Surf Weekly로 우수 사례를 전사에 공유한다.
- 이 시트와 연결: 섹션 4 집중도. 확산 담당자를 구조적으로 지정하면 신호 A(단독 운영)가 한 사람에게 몰리는 상태에서 벗어나기 시작한다.
- 출처: toss.tech/article/ai-surf-day (토스 자사 기술블로그)

2 Shopify — 공개 채널만 쓰게 강제한다

- 무엇을 했나: 사내 에이전트 River를 DM과 비공개 그룹에서 못 쓰게 막고 공개 Slack 채널에서만 동작하게 설계했다. 최근 30일 기준 세션 59,918건, 관여 임직원 7,000명 이상(자사 블로그가 공개한 수치).
- 이 시트와 연결: 섹션 4 집중도. 문서화 대신 관찰 가능성 자체를 강제해 좋은 사용법이 전파되게 만든 반대 접근이다.
- 출처: shopify.engineering/under-the-river (Shopify 자사 엔지니어링 블로그)

3 당근 — 만든 사람이 곧 쓰는 사람이다

- 무엇을 했나: 비개발자 구성원이 프롬프트만으로 기능을 만드는 사내 툴 프롬프트 스튜디오로 슬랙봇을 하루 만에 구현했고, 아이디어를 낸 본인이 직접 개선에도 참여했다.
- 이 시트와 연결: 섹션 4 집중도. 고칠 수 있는 사람을 개발자에서 비개발자까지 넓혀서, 신호 B(대체 운영 가능자 없음)가 생길 자리를 줄인 쪽이다.
- 출처: medium.com/daangn — 당근 AI Show & Tell #1 (당근 자사 기술블로그)

4 GitLab — 프롬프트를 릴리스와 분리해 배포한다

- 무엇을 했나: AI Gateway 안에 Agent Registry를 두어 프롬프트와 모델 설정을 YAML로 분리했다. 프롬프트를 고치거나 모델을 바꿀 때 GitLab 본체 릴리스 없이 반영된다.
- 이 시트와 연결: 섹션 2 귀속. 무엇이 언제 배포됐는지 추적 가능한 구조라, 비용이 누구 업무에 붙는지도 같이 따라간다.
- 출처: 공식 블로그 텍스트 자료는 확인하지 못했고, 자사 컨퍼런스 발표 영상을 정리한 ZenML LLMOps DB 기반이다.

5 Shift — 겪은 것 / 한 것 / 아직 안 된 것

우리가 겪은 것

- 자체 실측 결과 AI 실행 기록의 **46%**만 사람과 업무에 귀속됐다. 나머지 54%는 돈은 정확히 나갔고 토큰 수도 맞는데 누구 업무인지가 비어 있었다.
- 과거 데이터는 복구되지 않는다. 귀속은 나중에 붙이면 된다고 미루는 항목이고, 미루는 순간 그 구간은 영원히 비어 있다.
- **이 시트와 연결:** 섹션 2 귀속.

버전 — 우리가 한 것

- 사내 에이전트 팩에 버전을 붙이고 레지스트리에 발행한다. 버전 번호를 네 곳에서 동시에 올려야 발행이 되고, 하나라도 빠지면 거부된다. 사본이 조용히 갈라지는 것을 막으려고 일부러 뻑뻑하게 만든 장치다.
- 팩 파일은 내용 해시로 저장한다.
- **이 시트와 연결:** 섹션 4 집중도.

공용 런타임 — 우리가 한 것

- 팩은 개인 기계가 아니라 공용 런타임에서 실행된다. 같은 팩이면 누가 돌려도 같은 환경에서 돈다.
- 팩 코드는 표준 라이브러리와 주입된 실행 컨텍스트만 쓸 수 있다. 호스트의 무거운 자원은 등록된 이름으로만 호출한다. 실행 환경이 사람마다 달라지는 경로를 없앤 것이다.
- **이 시트와 연결:** 섹션 4 집중도.

아직 안 된 것

- 발행된 팩의 코드는 런타임이 실행을 거부한다. 내용 해시는 전송 중 변조만 막을 뿐 발행자를 신뢰할 근거가 아니기 때문이다. 실행되는 코드는 배포 이미지에 담긴 것뿐이고, 이 제약을 풀려면 발행자 서명 검증이나 실행 샌드박스를 먼저 만들어야 한다. 아직 안 만들었다.
- 감사 기록에 위임 체인이 없다. 행위자가 문자열 하나라서 누가 시켜서 이 작업이 났는지를 되짚을 수 없다.
- 호출 전에 예산을 자동으로 끊는 기능은 아직 라이브가 아니다.