

Schift.

AX 2년차 플레이북

구독은 했는데 아무도 말 안 해준 다섯 가지.

2026년 8월 20일 웨비나에서 다룬 내용을 조직에 공유할 수 있는 형태로
정리했습니다.

주식회사 룸821

2026년 8월 · contact-us@room821.im

구독은 했는데 아무도 말 안 해준 다섯 가지

AI를 몰라서 생긴 문제가 아닙니다. 도구를 들고 교육까지 마친 조직에서만 나타나는 문제입니다.

1 모델 단가는 내려가는데 청구서는 오른다

단가 인하보다 쓰는 업무와 실행 횟수가 더 빨리 늘어납니다. 총액만 보면 원인을 지목할 수 없습니다.

2 AI를 어디에 썼는지 설명할 ROI가 없다

청구서에는 모델 이름과 금액만 남습니다. 그 소비가 어떤 업무의 것이었는지는 아무 데도 적히지 않습니다.

3 회사에서 안 주면 승인 밖에서 쓴다

차단은 사용을 없애지 않고 기록만 없앱니다. 남는 것은 통제가 아니라 사각지대입니다.

4 잘하는 직원의 결과가 다음 사람에게 안 넘어간다

A가 만든 것은 A의 계정에만 남습니다. B는 처음부터 다시 시작하고, 같은 요청에 토큰이 두 번 나갑니다.

5 같은 정답이 나와도 비용은 다를 수 있다

정확도가 같은데 건당 비용이 몇 배씩 갑니다. 총액으로는 이 차이가 보이지 않습니다.

다섯 가지는 서로 다른 문제가 아닙니다. 전부 「남는 기록이 없다」는 한 가지 결론입니다.

그래서 이 보고서는 도구를 비교하지 않습니다. 어떤 모델을 쓰든 답이 달라지지 않는 항목만 다룹니다. 각 장은 **지금 무엇이 보이지 않는가**와 **무엇을 남기면 보이기 시작하는가** 두 부분으로 되어 있습니다.

읽는 순서는 앞에서부터가 맞지만, 지금 가장 아픈 장부터 펴셔도 됩니다. 다섯 장이 각각 독립적으로 읽히도록 썼습니다. 마지막 세 장은 그래서 무엇을 남길지, 30일 동안 무엇을 할지, 어디서부터 시작할지를 다룹니다.

숫자는 전부 출처를 각주에 적었습니다. 표시가 없는 것은 자사 운영 기준이거나 본 웨비나 사전 설문 결과입니다.

왜 지금 이 질문이 나오는가

AI 비용을 보는 일은 앞서가는 게 아니라 **기본값**이 됐다

2년 만에 표준이 바뀌었습니다. 지금 안 보고 있으면 뒤쳐진 쪽입니다.

31%

2024

AI 지출을 비용 관리 대상으로

63%

2025

1년 만에 두 배

98%

2026

사실상 전부

FinOps Foundation 조사에서 AI 지출을 비용 관리 대상으로 올린 조직 비율은 2024년 31%, 2025년 63%, 2026년 **98%**입니다. 불과 2년 만에 사실상 전부가 되었습니다.

도구도 따라왔습니다. 팀 단위로 비용을 귀속하고, 사용자별 예산과 공용 크레딧 풀로 사용 범위를 제어하는 기능이 이미 표준 제품에 들어 있습니다. **누가 얼마나 썼는지는 이제 답이 나옵니다.**

답이 안 나오는 건 그다음입니다

어떤 업무를 어떤 버전으로 끝냈는지, 그 결과가 실제로 채택돼 다시 쓰였는지. 청구서에는 모델 이름과 금액만 남고, 그 소비가 성과였는지 낭비였는지는 판단해주지 않습니다.

5배

비용 가시성의 ROI 확립 용이성

비용이 전부 보이는 집단 15% vs 나머지 3%

88% vs 7%

쓰는 조직 vs 설명되는 투자

정기 사용 88% · ROI 확립 7%

한 업무 기능 이상에서 AI를 정기적으로 쓴다는 응답은 88%인데, 투자를 정당화할 만큼 ROI가 확립됐다는 응답은 7%입니다. 그런데 **비용이 전부 보이고 모니터링된다고 답한 집단에서는 ROI 확립 응답이 15%로, 나머지 집단의 3%보다 5배 높았습니다.**

순서가 여기서 갈립니다. ROI를 먼저 증명하고 비용을 보는 게 아니라, **비용이 보이는 조직이 ROI를 말할 수 있게 됩니다.**

하나

모델 단가는 내려가는데 청구서는 오른다

단가가 아니라 호출량과 재시도가 총액을 정합니다.

저가형 모델의 단가 압박은 최근 한 차례 꺾였습니다. 그런데 단가가 내려가도 AI를 쓰는 업무가 늘고, 실행이 길어지고, 복잡한 요청이 고가 모델로 물리면서 **전체 토큰 소모량과 소모 금액은 다시 올라갑니다.**

호출량 × 건당 토큰 × 재시도 증가 → 토큰 총액의 증가

30% → 61%

월 100억 토큰 이상
2026 → 2028 전망

7% → 23%

월 1,000억 토큰 이상
같은 기간

24배

2030년 토큰 소비량
현재 추정 글로벌 용량 대비

Goldman Sachs는 2030년 토큰 소비량이 현재 추정 글로벌 용량의 24배를 넘을 수 있다고 봤습니다. **소비가 그만큼 커지면 총액으로는 판단할 수 없게 됩니다.**

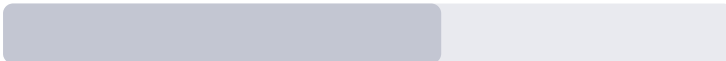
예산 초과는 이미 거의 전부가 겪었습니다

95%



최소 하나에서 예산 초과

60%



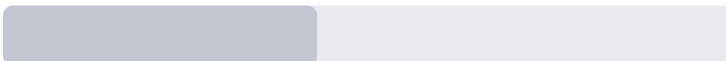
데이터·컴퓨팅 (전년 64%)

55%



지원 SW (전년 61%)

43%



모델 (전년 48%)

50%



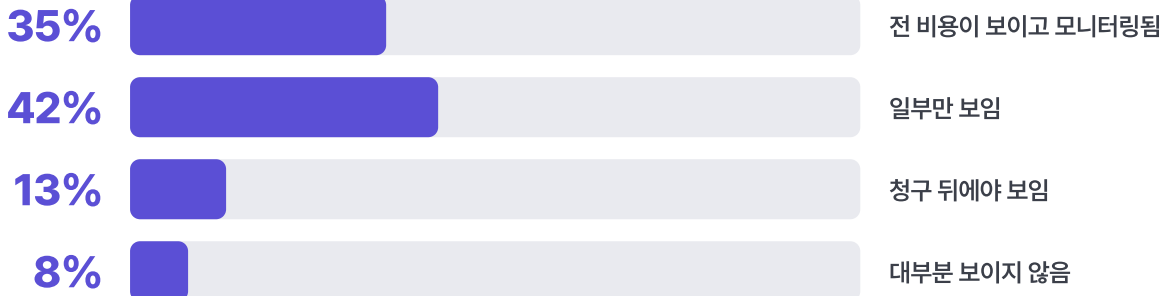
인재 (전년 48%) — 유일하게 상승

응답자의 95%가 최소 하나에서 예산 초과를 겪었습니다. 데이터, 지원 SW, 모델은 초과 경험이 모두 줄었습니다. 인프라와 도구 비용은 한 해를 겪고 나니 예측이 불기 시작했다는 뜻입니다.

인재만 올랐습니다. 유일하게 반대 방향입니다. 도구를 늘릴수록 그 도구를 붙잡고 있어야 할 사람이 같이 늘어나는데, 그 인건비는 어느 청구서에도 안 찍힙니다. 에이전트를 누가 만들었고 누가 다시 손보고 있는지가 기록으로 남지 않기 때문입니다.

AI를 어디에 썼는지 설명할 ROI가 없다

비용이 전부 보이는 조직은 세 곳 중 한 곳입니다.



비용이 전부 보이고 적극적으로 모니터링된다는 응답은 35%입니다. 나머지는 일부만 보이거나, 청구서가 나온 뒤에야 보이거나, 대부분 보이지 않습니다. 쓰는 조직은 아홉 곳 중 여덟, 투자를 설명할 수 있는 조직은 열넷 중 하나입니다.

다음 3년의 질문은 '얼마나 쓰나'가 아니라 '누가 무엇에 썼나'입니다

3.6배

IT가 단독으로 정하는 빈도

IT 리더십 51% vs 사업·기능 조직 14%

97%

3년 안에 확장할 수 있다

그 자신감이 이미 예산으로 확정되는 중

이 소비를 정하는 자리는 IT 리더십이 51%이고, 실제로 AI를 쓰는 사업 및 기능 조직은 14%입니다. 쓰는 사람과 정하는 사람이 갈라져 있으면, 새는 토큰과 성과를 낸 토큰을 구분할 원장이 어디에도 남지 않습니다.

그래서 무엇을 남겨야 하나

층위	확인할 질문	남길 것
도구	무엇을 구독하고 결제하고 있는가	도구, 결제 방식, 월 단가
사용	누가 얼마나 사용하는가	사용자, 토큰, 실행 횟수
업무	어떤 결과를 만들기 위해 썼는가	업무, 산출물, 품질, 비용

빠져 있는 건 대부분 세 번째 줄입니다. 도구와 사용은 이미 도구가 세어 줍니다.

회사에서 안 주면 승인 밖에서 쓴다

차단이 사용을 없애지 않습니다. 기록만 없앱니다.

81%

승인하지 않은 AI 도구를 쓰는 직원

45%는 차단을 우회한다고 답변

32%

에이전트를 프로덕션에 올린 조직

12개월 내 도입 가능성 높음 25%

직원 응답자의 81%가 승인하지 않은 AI 도구를 쓴다고 답했습니다. 45%는 차단을 우회한다고 답했습니다. 같은 시기, 에이전트를 실제 프로덕션에 올린 조직은 32%로, 12개월 내 도입 가능성이 높다고 답한 25%를 더하면 1년 안에 절반 이상이 올라옵니다. 아예 계획이 없다는 응답은 4%뿐입니다.

승인 밖 사용은 규율의 문제가 아니라 공식 경로의 기능과 승인 속도의 문제입니다

공식 경로가 느리면 사용이 사라지는 게 아니라 기록이 사라집니다. 그 상태로 1년이 지나면, 지금 승인 밖에서 굴러가던 것들이 **검증 없이 프로덕션으로 승격됩니다**. 1년 뒤에 문제가 되는 건 오늘 차단한 도구가 아니라, 차단당한 채로 쓰이면서 아무 기록도 남기지 않은 그 사용입니다.

잡아야 할 것은 사용이 아니라, 기록만 유실되는 부분입니다.

다음 투자에서 모델보다 지원 계층이 먼저 언급됐습니다



투자 확대가 필요하다는 응답에서 지원 SW가 61%로 가장 앞이고 모델이 45%로 가장 뒤입니다. 운영을 보이게 하는 계층도 투자의 일부라는 뜻입니다.

잘하는 직원의 결과가 다음 사람에게 안 넘어간다

사전 설문에서 가장 많이 꼽힌 문제입니다. 응답자의 47%.

지금



바뀌면



같은 요청에 토큰이 두 번 나가지만, 쌓이는 자산은 없다

같은 산출물을 A와 B가 각각 만들면 결과가 갈립니다. 프롬프트도, 참고 자료도, 품질 기준도 **A의 머릿속과 A의 계정에만 있기 때문입니다**. B는 처음부터 다시 시작하고, 같은 요청에 같은 토큰이 두 번 나갑니다.

사전 설문에서 가장 많이 꼽힌 두 가지가 「A가 만든 걸 B가 그대로 쓰고 싶다」(47%)와 「일이 잘 쓰는 사람에게 몰린다」(24%)였습니다. **둘은 같은 문제의 앞뒤입니다**. 공용 버전이 없으니 요청이 잘 쓰는 사람에게 몰리고, 그 사람이 만든 건 그 사람에게만 남습니다.

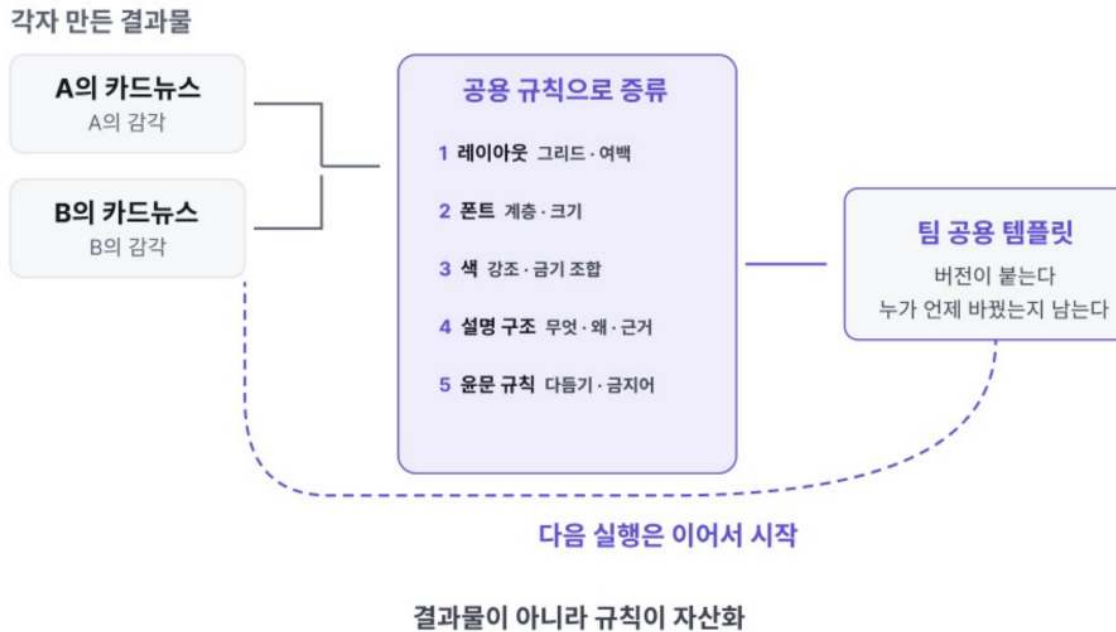
요청 창구가 한 사람이면, 그 사람이 나갈 때 요청도 같이 사라집니다

병목은 사람이 아니라 설계에서 생깁니다. 본인이 쓰는 에이전트에 도구 순서, 신뢰 경계, 롤백 요령을 넣어 두면 예외 처리까지 기록됩니다. 그걸 공용 버전으로 올리면 다음 사람이 이어서 실행합니다. **병목의 해소는 발행된 단위를 다른 사람이 실행하게 만드는 일입니다**.

MLOps, AIOps, AgentOps 엔지니어를 아직 확보하지 못해 추후 확충하겠다고 답한 조직이 27%입니다. 도구보다 운영 체계가 늦습니다.

결과물이 아니라 규칙이 자산이 된다

파일을 공유하면 파일 공유입니다. 자산이 되려면 판단이 같이 남아야 합니다.



차이를 취향으로 두면 개인 도구만 늘어난다

같은 업무인데 결과가 다르면 보통 「누가 더 잘한다」로 끝납니다. 그러면 남는 건 잘하는 사람 한 명이고 조직에는 아무것도 안 쌓입니다. A의 결과에서 규칙을 뽑아 팀 공용 템플릿으로 올리면, B의 다음 실행이 조직 자산에서 시작합니다. 공유되는 건 산출물이 아니라 **산출물을 만드는 기준**입니다.

제일 중요한 데이터는 반려 데이터입니다

공용 버전을 올려놓기만 하면 그건 파일 공유입니다. 자산이 되려면 무엇이 맞는 답이었는지가 같이 남아야 합니다.

반려할 때 고르는 **사유 선택지가 곧 분류 체계**이고, 그게 쌓이면 다음 버전이 무엇을 피해야 하는지 알게 됩니다. 승인만 기록하면 절반만 남습니다.

단계	하는 일	남는 것
1. 에이전트 초안	반복 처리의 자리	버전, 실행 경로
2. 사람 검수	직무가 판단으로 승격	승인 기록, 반려 사유
3. 다음 버전	사유가 그대로 판단 기준화	분류 체계, 회귀 기록

다섯

같은 정답이 나와도 비용은 다르다

청구서에는 안 나타납니다. 부족과 낭비는 업무 단위에서만 확인됩니다.

1.25원

일반 경로
검색 없음

0.16원

저가 모델 + 검색 + 톨콜링
분류는 모델, 계산은 코드

자사 경리 업무 검증에서 정답 스코어가 동일한데 건당 비용이 약 8배 차이가 났습니다. 분류만 모델이 하고 산술은 코드가 하는 구성입니다. 같은 결과가 나온다면 건당 비용이 낮은 쪽으로 가야 합니다.

예산 초과 원인은 둘로 갈립니다

구분	무엇인가	어디서 생기는가
부족	한도와 용량이 모자란 경우	정보 부족으로 인한 추가 결제, AI 런타임 용량 부족
낭비	같은 결과에 더 쓴 경우	같은 일을 두 번 실행, 상위 모델 불필요 사용, 재시도, 불필요하게 긴 컨텍스트

맞은 답과 끝낸 일을 기준으로 일 건당 비용을 보면 둘을 나눌 수 있습니다. 총액으로는 부족인지 낭비인지 구분되지 않습니다.

검색 품질을 올리면 찾는 시간과 근거 없는 답이 함께 줄었습니다

25.0%

검색 도구 없음
hit@10

75.0%

검색 도구 사용
hit@10

11.6% →
0%

할루시네이션
찾는 시간 177초 → 91초

문서가 70배 늘어난 위키 데모에서 검색 도구를 붙이자 hit@10이 25%에서 75%가 됐습니다. 재시도와 불필요한 컨텍스트가 줄어, 비용도 업무 단위에서 다시 볼 수 있습니다.

그래서 무엇을 남겨야 하나

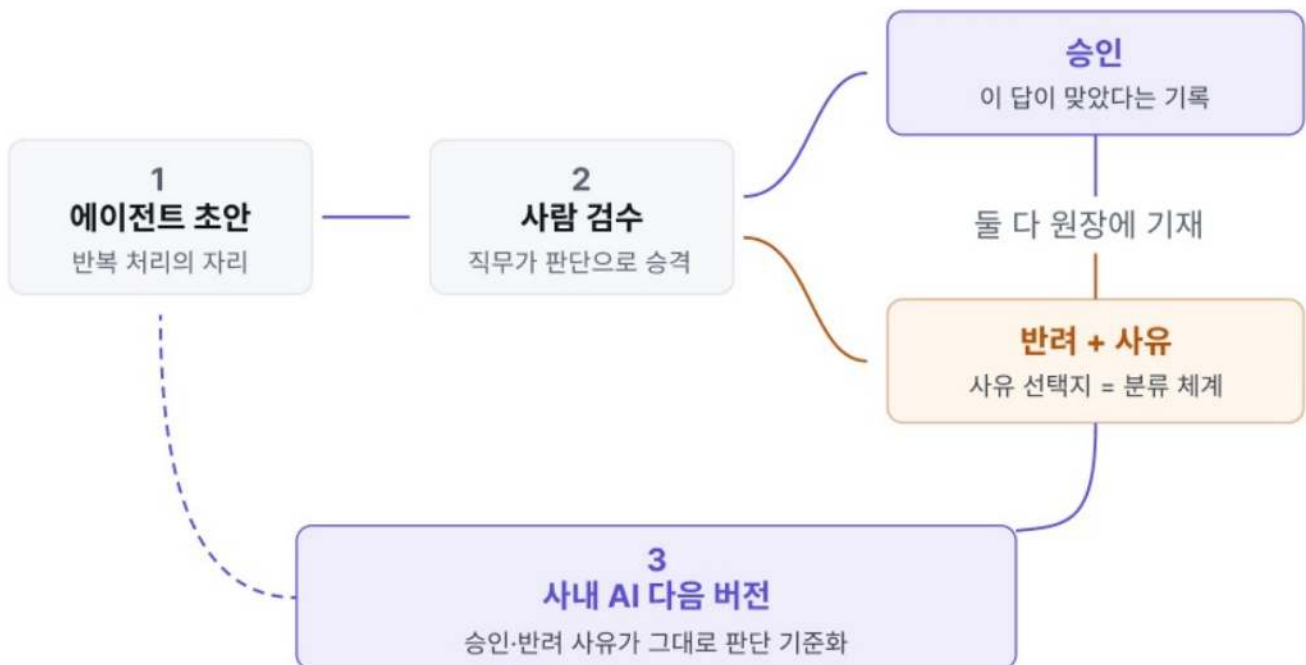
원장에 남아야 할 최소 세 줄

거버넌스는 개입, 비용, 결과를 다음 판단에 연결하는 운영 규칙입니다.

운영 기능	질문	남겨야 할 것
Govern, Map	누가 어떤 업무를 맡나	담당, 버전, 경로
Measure	얼마로 끝냈나	비용, 품질, 성공 라벨
Manage	언제 멈추고 되돌리나	반려, 개입, 회귀 기록

토큰이 늘면 비용을 줄이기 전에 무엇이 실제 업무에 쓰였는지부터 다시 묻게 됩니다. 지금까지 본 자금 구멍 — 청구서에 AI 금액만 남고, 사람 비용은 안 찍히고, 승인 밖 사용은 아예 기록이 없는 것 — 이 셋은 전부 같은 결론입니다. 남는 것이 없다는 것.

비용을 이해하는 책임이 매우 명확하다는 응답이 34%, 사람의 개입 지점이 매우 명확하다는 응답이 35%였습니다. **셋 중 두 곳은 총액만 있고 그 총액을 업무로 되짚을 축이 없습니다.** 비용을 줄이는 일은 이 세 줄이 남은 다음에 시작됩니다.



승인과 반려가 둘 다 원장에 기재돼야 다음 버전의 기준이 된다

회사 전체가 아니라 한 팀의 한 업무부터

전사 정책을 한 번에 만드는 일이 아닙니다. 한 반복 업무를 30일 동안 기록하고, 비교하고, 공용화하는 일입니다.



쓰던 도구는 그대로 둔다. 기록이 모이는 곳만 하나로 만든다

1 관측 — 흩어진 사용 기록을 한 곳에 모은다

쓰던 도구는 그대로 둡니다. 업무 분류, 질의셋, 비용이 한 원장에 모이는 것만 만듭니다.

2 제안 — 공통 업무를 찾아 부서 공용 초안을 만든다

여러 사람이 같은 요청을 반복하고 있는 지점이 먼저 드러납니다. 거기가 첫 후보입니다.

3 수렴 — 사람마다 다른 방식을 한 벌로 합친다

차이를 취향으로 두지 않고 규칙으로 꺼냅니다. 30분 미팅 한 번이면 대부분 정리됩니다.

4 확정 — 회사가 정한 업무 방식 한 벌로 발행한다

각자 쓰던 AI에서 그대로 호출합니다. 신규 입사자는 첫날부터 같은 방식에서 시작합니다.

완료했을 때 남아야 하는 것

사용과 비용 인벤토리, 업무별 귀속과 건당 비용, 승인과 반려 기록, 버전이 붙은 공용 자산과 대체 운영자. **이 넷이 남으면 30일은 성공입니다.** 숫자를 모르면 추정하지 말고 모름으로 남기십시오. 모름은 실패가 아니라 다음 확인 작업의 이름입니다.

AI 비용 거버넌스 PoC, 30분 미팅으로 시작

별도로 준비하실 내용 없이, 지금 상태 그대로 보면서 범위만 정합니다.



PoC 문의 · 30분 미팅
schift.io/consultation/call

1 AI 지출 목록과 미귀속 항목을 함께

합계가 처음 나온 자리에서 대부분 놀랍니다. 정확하지 않아도 괜찮습니다.

2 부담 없이 시작할 한 팀, 한 업무를 선정

반복되고, 결과를 판단할 수 있고, 담당이 분명한 업무가 좋습니다.

3 해당 범위로 PoC를 어디서부터 할지

그 자리에서 결론이 안 나도 괜찮습니다. 범위만 좁혀도 성과입니다.

아직 미팅이 이르다면

AI 전환 자가진단 — 우리 조직이 어디쯤인가

schift.io/ai-transition-diagnosis

연락처 없이 바로 보실 수 있습니다. 몇 가지 질문에 답하면 지금 걸림돌과 가장 약한 축이 나옵니다. 뒤에 붙은 진단 시트를 먼저 채워 보셔도 됩니다.

비용을 줄이는 일은 세 줄이 남은 다음에 시작됩니다.

담당과 버전과 경로. 비용과 품질과 성공 라벨. 반려와 개입과 회귀 기록. 이 셋이 원장에 남으면, 그때부터 무엇을 줄일지 말할 수 있습니다.

주식회사 룸821

contact-us@room821.im

schift.io

2026년 8월 · AX 2년차 플레이북

AI를 도입한 지 1~2년 지난 조직이 지금 답해야 할 질문 네 가지를, 우리 회사 숫자로 채워서 답으로 바꾸는 진단 시트입니다. 비용을 세고, 사람·업무에 붙이고, 반복되는 일을 공용화할 기준을 만듭니다.

작성자		작성일	
대상 조직		기준 기간	

★ 경영진 요약

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

지표	값	해석·주의
미귀속 비율 (%)		
1인당 실효 부담 (원/월)		
채용 비용 기준 환산 규모 (명)		
병목 신호 (2개 이상 겹침 항목 수)		

그래서 무엇을 결재·요청하는가

요청 내용

실행 범위 (한 팀·한 업무)

결재 대상 (예산·인력·도구·정책)

기록 항목 (담당자·업무·비용·버전)

실행 책임자·완료 기한

필요 자원 (예산·인력)

목표 지표 (현재 → 목표)·다음 점검일

예: 다음 분기에는 한 팀·한 업무의 사용·지출·결과를 연결하고 귀속률 X%를 확인한다

Schift도 같은 질문을 내부에서 확인하는 중입니다

AI 사용과 지출을 먼저 세고, 누가 어떤 업무에 사용했는지 연결한 뒤, 다시 실행할 만한 결과를 공용 버전으로 남깁니다.
아직 모든 실행이 귀속되거나 공용화된 상태는 아닙니다. 그래서 이 시트처럼 기록·귀속·공용화가 어디까지 되었는지를 숫자와 업무 단위로 계속 확인하고 있습니다.

AX 2년차 진단 시트 · 초대

작성자 전용 30분 콜

시트 내용을 보내 주세요. 확인 후 가능한 시간을 따로 제안드립니다.



QR 스캔
콜 신청

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	
		귀속 / 미귀속	
		귀속 / 미귀속	
		귀속 / 미귀속	
		귀속 / 미귀속	
합계			

계산식 · 판정 기준 → 5쪽 부록 B

3 손익분기 — AI 지출이 채용 비용과 같은 규모인가

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 비용 기준 환산 규모 (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:F11)
- 전사 연 합계 = 전사 월 합계 × 12 =합계셀*12
- 1인당 실효 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 비용 기준 환산 규모 = 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 — 프롬프트를 전사 저장소로 올린다

- 무엇을 했나: LLM Router(비용·키 중앙화) · Prompt Studio(프롬프트 버전·배포·MCP Hub) · KarrotChat(에이전트 발견·사용). Prompt Studio가 전사 AI 지식 저장소가 되고, 만든 에이전트가 다른 팀으로 전파된다.
- 이 시트와 연결: 섹션 2 귀속(Router로 서비스별 비용) · 섹션 4 집중도(개인 프롬프트가 공용 자산으로 올라가는 자리).
- 출처: medium.com/daangn — 당근의 GenAI 플랫폼 (자사 기술블로그, 2025-12)

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: ZenML LLMOps DB가 정리한 GitLab 컨퍼런스 발표 내용(2차 자료).