

기술

AI에게 준 규칙, 무엇을 보고 지워야 할까요

방동걸 (주)웨이스 대표이사

2026년 9월 6일

AI에게 일을 시키려면 규칙을 적어 줘야 합니다. 무엇을 먼저 확인하고, 무엇을 하면 안 되고, 어디서 멈춰 사람에게 물어야 하는지를 문서로 만들어 매번 읽힙니다. 문제는 그 문서가 한 방향으로만 자란다는 것입니다. 사고가 날 때마다 조항이 하나씩 붙고, 지우는 절차는 아무도 만들지 않습니다. 웨이스도 VEXPLOR 개발과 사내 업무에서 같은 문서를 매일 읽히고 있어, 어디부터 지울지 따져 봤습니다.

저희도 그렇게 몇 달을 쌓았습니다. 그래서 어느 날 조항 두 개를 지우자고 제안했고, 다섯 관점으로 나눠 검토했습니다. 결과는 이렇습니다.

- 제안 2건 중 **1건이 만장일치로** 기각됐습니다.
- 제가 근거로 든 주장 **7개가 실측과 달랐습니다**.
- 그 조항이 지정한 실행 방법의 기록은 **저희 문서 전체에서 0건**이었습니다.
- 실행 7건은 **전부 조항과 다르게 한 것**이었습니다.

지우는 일이 이렇게 어긋난 이유와, 그래서 만든 판별 기준을 적습니다.

01

쌓기만 하는 규칙이 왜 문제인지는 이미 측정돼 있습니다

최근 몇 주 사이에 이 문제를 정면으로 다룬 연구가 나왔습니다. 둘 다 아직 심사를 통과하지 않은 프리프린트라 수치를 근거로 단독 인용하지는 않고, 문제를 세우는 재료로만 가져옵니다.

한 연구는 에이전트에게 기억 저장소를 주고 읽고 쓰는 시점 자체를 학습시켰습니다. 기억을 한 번 조회하는 것이 실제로 한 걸음 움직이는 것과 같은 비용이 들도록 했더니, 학습이 진행될수록 에이전트는 저장소를 덜 불렀습니다. 저장소도 계속 커지지 않고 병합과 정리를 거쳐 작아졌습니다. 흥미로운 것은 행동마다 사라지는 속도가 달랐다는 점입니다 — 과거 경험을 찾아보는 조회가 가장 오래 값어치를 유지했고, 새로 적어 두는 행동은 빠르게 사라졌습니다. 다만 논문은 이 분포가 환경에 따라 달라진다고 못박았습니다(arXiv 2608.05446).

다른 연구는 반대쪽을 켜습니다. 모델에게 자기가 쓸 실행 환경을 스스로 만들고 개선하게 하고 2,207개 비공개 과제로 채점했습니다(arXiv 2609.01437). 결과는 냉정했습니다. 코드·검색 영역에서 사람이 만든

기준선에 크게 못 미쳤고, 반복 개선으로 얻은 이득은 불안정했으며, 실행 모델을 바꾸면 그 이득이 잘 옮겨가지 않았습니다.

두 결과를 붙이면 이렇게 읽힙니다. 규칙은 줄여야 하지만, 줄이는 판단을 규칙을 쓰는 쪽에 맡기면 안 된다. 저희가 정확히 그 자리에서 넘어졌습니다.

02

손보자고 올린 조항은 두 개였습니다

첫째는 여러 명이 나눠 일할 때 결과를 대화창 답변이 아니라 파일로 받는다는 조항입니다. 답변이 안 와도 파일이 있으면 완료, 답변이 와도 파일이 없으면 미완료로 판정합니다.

둘째는 되돌리기 어려운 판단은 세 번 따로 계산한 뒤 비교한다는 조항입니다. 견적, 사업 진출, 자본 구조처럼 틀리면 되돌리기 어려운 결정이 대상이었습니다.

손보자고 본 근거는 단순했습니다. 첫째 조항은 "요즘 그런 사고가 안 나니 폐기 조건이 발화한 것 아닌가", 둘째 조항은 "적용 대상이 너무 넓어 실제로는 면제만 반복된다"였습니다. 둘 다 그럴듯했고, 둘 다 틀렸습니다.

03

먼저 숫자가 틀렸습니다

검토를 나눠 맡기자 제일 먼저 무너진 것이 제 집계였습니다.

제가 적은 것	실제로 세어 보니
면제 18회 대 실행 1회	실행 7 · 면제 6 · 미실행 7
그 방법을 센 기록이 없다	기록이 있고 1회에서 멈춰 있었다
규칙이 면제로만 소비된다	면제 6건은 전부 정확한 면제였다
지키는 비용이 3배다	실행 7건은 전부 병렬 처리였다

원인은 읽는 방식이었습니다. 저희 기록 원장은 최근 것이 위로 올라오는데, 저는 그것을 시간순으로 읽고 "미이행"이라고 적힌 줄에서 멈췄습니다. 같은 날 아래쪽에 완료 기록이 있었습니다. 간판으로 삼았던 실패 사례가 실은 같은 날 끝난 성공 사례였습니다.

여기서 배운 것은 규칙에 관한 것이 아니라 읽기에 관한 것입니다. 어떤 문서든 정렬 방향을 먼저 확인해야 하고, 상태를 담은 원장은 그 상태 칸까지 읽어야 답이 됩니다.

04

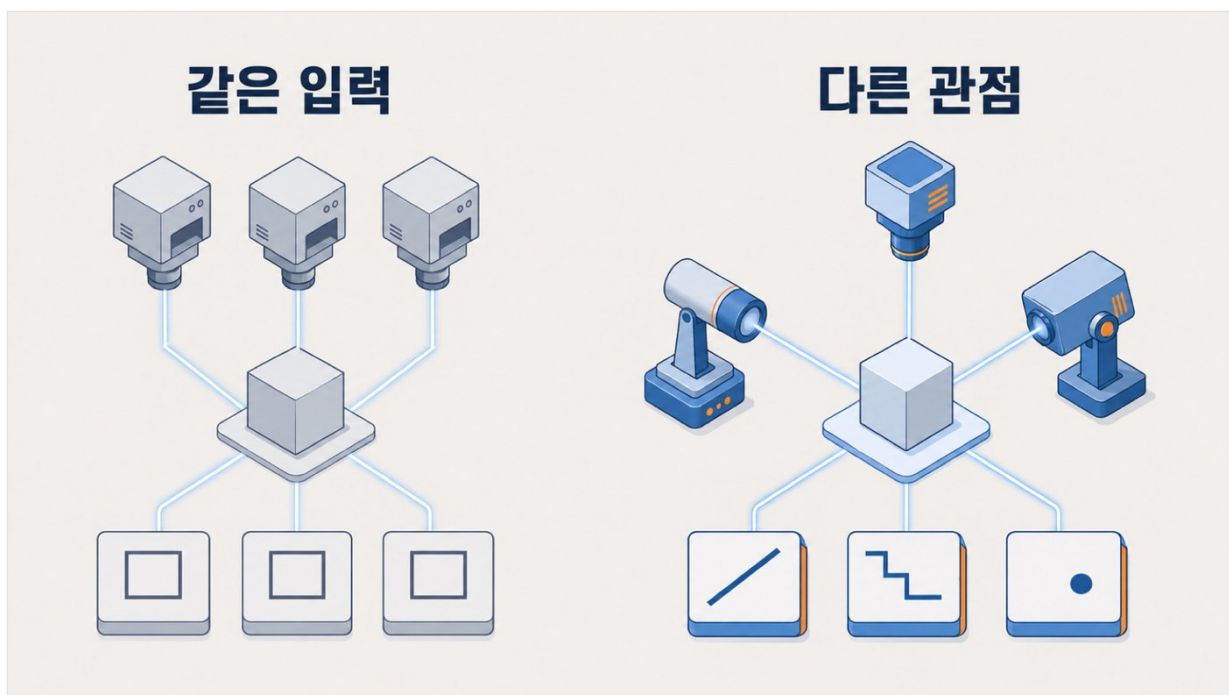
안 돌던 이유는 범위가 아니라 방법이었습니다

둘째 조항의 진짜 문제는 적용 범위가 아니었습니다. 조항이 지정한 **실행 방법이 아무도 쓸 수 없는** 것이었습니다.

조항은 "따로 연 대화창 세 개에서 각각 계산하라"고 적혀 있었습니다. 그런데 저희 기록 어디에도 그렇게 실행한 흔적이 없었습니다. 값이 실제로 나온 실행은 전부 조항과 다른 방식이었고, 두 건은 아예 "이 조항을 준용했다"고 적어 두기까지 했습니다. 조항을 지킨 사람이 0명인데 조항이 살아 있었던 것입니다.

무엇이 달랐는지 실행 기록 7건을 갈라 보니 선이 분명했습니다.

실행 방식	건수	결과
같은 질문을 그대로 반복	2	3판 전부 일치 · 검출 0
계산법·관점을 갈라 배치	5	4건에서 편차 검출



같은 질문을 세 번 반복하면 같은 답이 세 번 나올 뿐이고, 편차는 계산법과 관점을 갈라 배치했을 때 드러났습니다.

같은 입력을 세 번 넣으면 같은 답이 세 번 나옵니다. 안심할 근거가 되지 못할뿐더러, **같은 누락까지 세 번 복제**됩니다. 실제로 한 견적에서는 세 계산이 10% 안쪽으로 모였는데 여섯 항목을 똑같이 빠뜨렸고, 그것을 잡아낸 것은 반복이 아니라 그 분야를 아는 사람의 검토였습니다. **모이는 것은 정보가 아닙니다. 갈리는 것이 정보입니다.**

05

조항 둘이 서로를 부정한 채 18일 동안 놓여 있었습니다

방법을 조사하다 더 이상한 것을 찾았습니다. 그 실행 방법을 규칙에 넣은 날이 8월 18일인데, **이틀 뒤인 8월 20일에 다른 검토가 정확히 그 방법을 기각해 두었습니다.** 기각 사유는 이랬습니다 — 대화창을 따로 열어도 같은 모델이 같은 규칙 문서를 읽으므로 오류가 겹치는 정도가 줄지 않는다. 독립성은 창을 나누는 데서 오지 않고 결정적인 증거에서 온다.

기록 두 개가 서로를 부정한 채로 18일을 지냈습니다. 아무도 몰랐던 것이 아니라, **새 조항을 넣을 때 기존 판정과 대조하는 절차가 없었을 뿐입니다.** 규칙 문서가 커질수록 이런 자기모순은 늘어나는데, 문서는 커지기만 하니 대조 비용도 같이 커집니다. 조항을 넣는 쪽에 "이 방법을 이미 기각한 기록이 있는가"를 한 번 묻는 절차가 없으면, 문서는 계속 자기와 다투게 됩니다.

06

지우려던 조항의 반증 조건이 스스로를 충족하고 있었습니다

첫째 조항에는 "3회 돌려서 문제가 안 생기면 폐기한다"는 조건이 붙어 있었습니다. 검토 과정에서 이 조건 자체가 고장이라는 지적이 나왔습니다.

그 조항은 **완료의 정의를 바꾸는 규칙**입니다. 파일이 없으면 미완료라고 정의했으니, 규칙을 켜 둔 동안 "결과를 못 받은 사고"는 관측치가 언제나 0입니다. 0이 나오는 이유가 사고가 없어서인지, 규칙이 사고를 사고로 세지 않아서인지 결과만 보서는 구분되지 않습니다. 조건은 무엇을 해도 충족됩니다.



가드를 켜 둔 동안 관측치는 언제나 0이라, 결과만 보서는 사고가 없는 것인지 사고로 세지 않는 것인지 구분되지 않습니다.

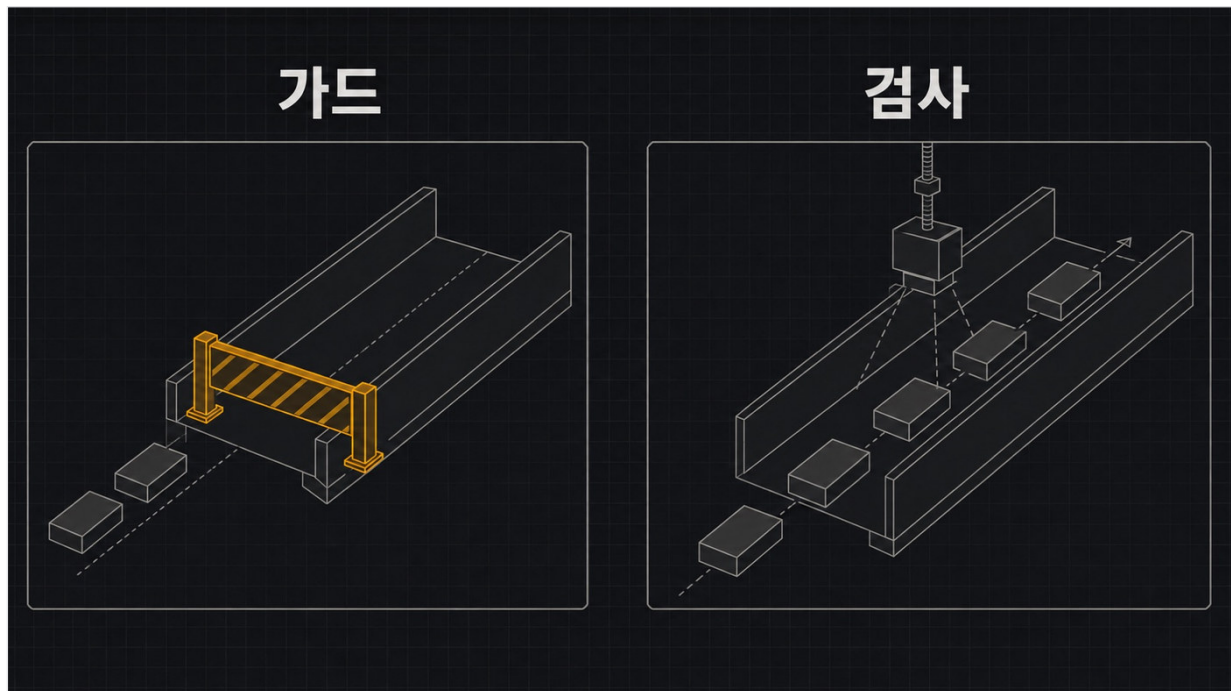
같은 고장이 제 축소안에도 있었습니다. "적용 범위를 좁히면 면제 비율이 좋아진다" — 당연합니다. 범위에서 빼면 면제가 아니게 되니까요. 행동이 아니라 정의로 충족되는 지표는 개선을 증명하지 못합니다.

07

그래서 규칙을 지울 때 이것부터 가릅니다

검토에서 나온 구분이 이번 결정의 실질이었습니다. 규칙에 조건을 붙일 때 먼저 통과시킵니다.

구분	하는 일	어떻게 폐기를 판정하나
가드	결함의 정의·발생 경로를 바꾼다	결과로는 안 된다 · 끄고 재 본다
검사	결함을 관측만 한다	검출률로 판정한다



가드는 결함이 생기는 경로를 막고 검사는 지나가는 것을 관측하므로, 둘 다 필요하되 폐기를 판정하는 방법이 서로 다릅니다.

판별 질문은 한 줄입니다.

"이 규칙을 끄면 그 문제가 다시 나는가, 아니면 문제는 원래 있는데 안 보이게 될 뿐인가."

앞이면 가드입니다. 증상이 없다는 것은 근거가 되지 않고, 되돌릴 수 있는 대상 하나에서 실제로 꺼 보거나, 그 문제를 만들던 조건이 사라졌다는 바깥 근거를 가져와야 합니다. 뒤면 검사입니다. 이걸 검출률로 판정이 됩니다.

규칙을 지우려 할 때 함께 보는 것 네 가지도 이번에 정리했습니다.

- 그 방법으로 실행된 기록이 실제로 있는가. 0건이면 범위가 아니라 방법이 문제입니다.
- 폐기 조건이 행동으로 충족되는가, 정의로 충족되는가. 후자면 조건부터 고쳐 적습니다.
- 안 돌아간 지점이 어디인가. 저희는 미이행 7건이 전부 "사람 결정으로 넘긴" 자리였습니다. 넘긴 뒤 아무 대기열에도 들어가지 않아 그대로 증발했습니다.
- 범위를 줄이면 무엇이 대상에서 빠지는가. 저희 축소안은 미이행을 해결하지 못한 채 대상에서 빼 버렸고, 잘 돌던 실행 3건까지 함께 뺐습니다. 미이행을 줄이는 게 아니라 미이행을 합법화하는 안이었습니다.

08

검사를 줄이자는 제안은 계산해 보면 대개 손해입니다

범위를 좁히자는 제 제안에는 품질 쪽 관점에서 반박이 하나 더 붙었습니다. 검사를 줄일지 말지는 취향이 아니라 계산으로 답이 나오는 문제라는 것입니다.

두 값만 있으면 됩니다. 검사가 실제로 결함을 잡아내는 비율과, 검사 한 번의 비용 대 결함이 뒤로 넘어갔을 때의 비용 비율입니다. 저희 실행 기록에서 앞의 값은 0.6이었습니다. 뒤의 값은 검토 한 번의 비용이 작고 틀린 건적 하나가 계약 전체를 흔드는 구조라 1,000분의 1 수준입니다. 앞의 값이 뒤의 값보다 세 자릿수 크면, 표본만 골라 검사하는 것이 최소 비용이 되는 구간은 없습니다. 전수 검사가 답입니다.

여기서 제 실수의 성격이 분명해졌습니다. 저는 "규칙이 잘 안 지켜진다"를 "규칙이 과하다"로 읽었습니다. 실제로는 규칙이 과한 게 아니라 지킬 방법이 없었던 것이고, 계산상 검사는 오히려 더 촘촘해야 했습니다.

09

판정 기준이 도구 권한에 흔들리는 자리도 나왔습니다

첫째 조항을 유지하기로 하면서 구멍 하나를 함께 막았습니다. 완료 판정 기준이 "일을 했는가"가 아니라 "파일이 있는가"인데, **파일이 있느냐 없느냐는 참여자의 도구 권한에도 좌우됩니다**. 읽기 전용으로 붙는 참여자는 파일을 만들 수 없어 결과를 답변으로 보냅니다.

사흘 동안 이 형태가 세 번 났고, 그중 한 번은 실제 손해로 끝났습니다. 검수 네 건 중 두 건이 파일을 못 만들었는데 진행자가 그것을 "결과 없음"으로 읽고 그대로 발행했고, 발행 직후에 사실 확인 지적 다섯 건이 나와 전 채널을 회수했습니다. 규칙은 정확히 작동했는데 **규칙이 전제한 도구 조건이 충족되지 않았던** 것입니다.

그래서 조항에 단서를 붙였습니다 — 읽기 전용 참여자의 답변은 진행자가 대신 저장하고 완료로 판정한다. 그리고 사람을 붙이기 전에 그 참여자가 파일을 만들 수 있는지 먼저 확인한다.

10

지운 것과 남긴 것

결과만 적으면 이렇습니다. 첫째 조항은 **영구 유지**로 바뀌었습니다. 폐기 조건을 댄 대신, 도구 권한이 없어 파일을 못 만드는 참여자가 있으면 진행자가 대신 저장하도록 단서를 붙였습니다. 둘째 조항은 범위를 그대로 두고 **방법을 교체**했습니다 — 관점을 갈라 병렬로 배치하고, 밖에 원본이 있는 값(벤더 견적·실판매가·규정 조문)은 애초에 대상이 아니라고 명시하고, 실행하지 않을 때는 대기 목록에 한 줄을 남기게 했습니다.

지운 조항은 하나도 없습니다. 대신 **조건 하나와 방법 하나가 죽었습니다**. 규칙 정리는 조항 수를 줄이는 일이 아니라, 실행할 수 없는 방법과 스스로를 충족하는 조건을 걷어내는 일이었습니다.

한 가지는 아직 못 했습니다. 검토 산출물을 담는 폴더가 17개·파일 85개까지 늘었는데 색인이 0입니다. 지우지 않는 규칙과 겹쳐서 계속 쌓입니다. 저장소가 커지는 문제는 앞의 연구가 병합과 정리로 다뤘던 바로 그 자리인데, 저희는 아직 손대지 못했습니다.

여러분의 규칙 문서에는 폐기 조건이 붙어 있나요. 그 조건은 행동으로 충족되나요, 아니면 정의로 충족되나요.

WACE(웨이스)는 공장이 스스로 판단하고 멈추지 않게 하는 제조 AI 운영체제 VEXPLOR를 만드는 회사다.



원문 blog.vexplor.com/post/which-rule-should-you-delete
발행 2026년 9월 6일 · 방동걸 (주)웨이스 대표이사

회사 주식회사 웨이스 (WACE Inc.) · 대표이사 방동걸 ·
사업자등록번호 439-87-02852
주소 (18462) 경기 화성시 동탄구 동탄영천로 150, 에이동(A동)
1702·1706호 (영천동, 실리콘밸리동탄)
연락처 전화 070-4730-0629 · contact@wace.me
홈페이지 wace.me · linkedin.com/company/wace-inc



원문



홈페이지



링크드인