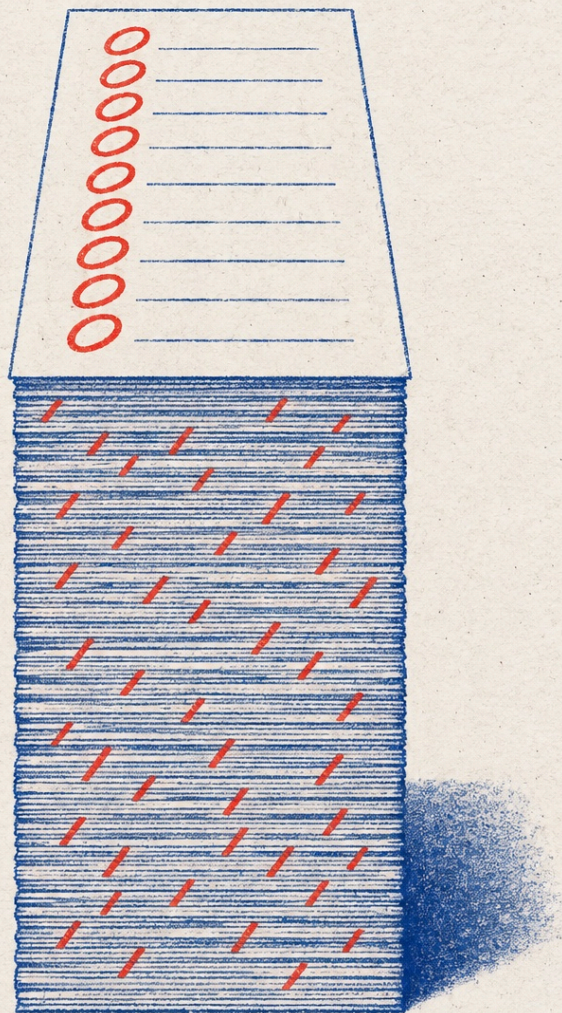


오답이 없는 연습

AI 엔지니어의 여섯 달



오답이 없는 연습

AI 엔지니어의 여섯 달

Zhuge Hyuk

2026

차례

1장 — 돌아간다

2장 — 열 번 중 열 번

3장 — 채점자를 켜다

4장 — 고친 자리

5장 — 삼십 퍼센트

6장 — 채점표가 거짓말할 때

7장 — 여섯 달

결론 — 마지막 줄

부록 A — 반증과 만료

부록 B — 여섯 달 계획

부록 C — 출처와 등급

돌아간다

7h53m 6m

대시보드의 계정 표 한 칸에 7h53m 6m 이 떠 있다. 토큰이 만료되기까지 7시간 53분, 마지막 갱신은 6분 전이라는 표시다. 아무도 쓰지 않아 그대로 두면 만료될 구독 계정의 토큰을, 화면 뒤에서 돌던 프로그램이 여덟 시간 가까이로 되살려 놓았다. 이 한 칸은 2026-06-13 오후 1시 39분(한국 시각) 공개 저장소 2lab-ai/llmux에 올라간 첫 커밋 속 연구 노트에 옮겨져 있다.¹

커밋 메시지는 한 줄이다. "chore: reset history around Rust implementation", Rust 구현을 기준으로 이력을 새로 시작한다는 뜻이다. 그 한 줄 밑으로 파일 60개, 24,993줄이 한꺼번에 들어온다. Rust 소스가 45개 파일에 20,169줄이고, 그 사이에 테스트 함수를 표시하는 속성(#[test]·#[tokio::test])이 311개 박혀 있다. 자동 빌드·배포용 워크플로 파일 셋(ci-preview·release)과 설계 문서 셋(스펙·구조·연구 노트)도 함께 들어온다. 이름은 아직 llmux가 아니라 teamagent, 판 번호는 0.1.0이다. 이 저장소는 내가 만들었다.²³

README의 첫 문장이 이 물건을 정의한다. "teamagent is a Rust local proxy for Claude Code." Claude Code는 Anthropic이 만든 터미널 코딩 에이전트이고, 프록시는 요청을 중간에서 받아 대신 보내 주는 프로그램이다. teamagent는 Claude Code가 보내는 요청을 같은 컴퓨터의 3456번 포트에서 받아, 여러 구독 계정 가운데 쿼터가 먼저 만료될 계정부터 쓴다. 첫날 요청이 실제로 나가는 길은 둘이다. Anthropic으로 가는 요청은 인증과 헤더를 고친 뒤 본문을 바이트 그대로 넘기고, OpenAI Codex 계정은 손으로 고르거나 넘칠 때 쓰는 예비로 붙인다. Gemini와 로컬 모델은 "컴파일만 확인한 빈 컵데기"라고 적혀 있다.⁴

연구 노트 첫 문단의 마지막 문장은 "Dogfooding on 2026-06-13"으로 시작한다. 만든 사람이 제 도구를 먼저 써 보는 일, 도그푸딩이 그날 Codex와 토큰 갱신, 대시보드 연결 방식, brew 배포에 대해 실제 백엔드에서 얻은 발견을 보냈다는 문장이다. 그 사용이 설계를 바꿨다. Claude Code 세션은 오래 살아 있는데 구독 계정의 OAuth 접근 토큰, 곧

로그인한 상태를 증명하는 출입증은 8시간 안팎이면 만료된다. 같은 커밋의 스펙은 이것을 도그푸딩 중에 나타난 두 번째 운영 문제라고 부르고, 그래서 v0.1은 창을 닫아도 뒤에서 도는 데몬을 기본으로 삼는("daemon-first") 로컬 프록시라고 적는다. 계정 표의 **v6m** 이 그 데몬이 한 일이다.⁵

그 오후의 기록은 짧은 간격으로 이어진다. 같은 날 나온 v0.1.0 릴리스의 태그가 이 커밋을 가리키고, 스펙 문서 머리에는 이미 "v0.1.0 shipped"라는 상태와 Homebrew 설치 경로가 적혀 있다. 3시 47분, README가 다시 쓰이며 첫머리에 표어 한 줄이 들어간다. "Models change every month. Your harness shouldn't." 모델은 달마다 바뀌어도 그 모델에게 일을 시키는 하네스, 곧 에이전트 도구와 설정의 묶음은 바뀌지 않아야 한다는 내기다. 4시 24분, 요청에 적힌 모델 이름을 보고 Claude 쪽으로 보낼지 Codex 쪽으로 보낼지 고르는 커밋이 들어간다. 5시 17분, 돌고 있는 데몬의 판이 새로 깔린 프로그램과 다르면 데몬을 스스로 다시 띄우는 커밋이 뒤따른다. 첫 공개 커밋에서 네 시간이 채 안 된 사이의 일이다.⁶

오후 한나절

그 커밋보다 꼭 3년 앞선 2023-06-13, OpenAI는 Chat Completions API에 함수 호출(function calling)을 넣었다. 개발자가 함수의 이름과 인자를 JSON Schema로 적어 건네면, 모델이 그 함수를 부를지 스스로 고르고 부를 때 넣을 인자를 JSON 객체로 내놓는다. JSON Schema는 필드마다 이름과 값의 종류를 정해 둔 명세다. 모델이 문장 대신 빈칸을 채운 서식을 돌려주기 시작한 셈이다. 대상은 그날 나온 gpt-4-0613과 gpt-3.5-turbo-0613이었다.⁷

같은 공지는 시작하는 법도 일렀다. 개발자 문서를 보고 시작하되, 함수 호출이 더 나아질 여지가 있는 사례를 찾으면 평가(evals)를 보태라고 했다. 같은 날 gpt-3.5-turbo의 입력가격을 25% 내리면서는 1달러에 대략 700쪽이라고 셈해 보였다.⁸

17일 뒤인 2023-06-30, Shawn Wang(swyx)이 Latent Space에 「The Rise of the AI Engineer」를 올렸다. 부제는 "창발하는 능력이 떠오르는 직함을 만들고 있다"로 시작한다. 둘째 문단에 이 문장이 있다. "2013년에는 5년과 연구팀이 있어야 해내던 넓은 범위의 AI 과제가, 2023년에는 API 문서와 남는 오후 한나절만 있으면 된다." 같은 글에서 그는 AI Engineer Summit을 연다고 알렸다.⁹

값 안의 실수

2024-05-30, Anthropic은 도구 사용(tool use)을 Claude 3 모델 전부에 정식으로 연다고 발표했다. 자사 Messages API뿐 아니라 Amazon Bedrock과 Google Cloud의 Vertex AI에서도 열었다. 도구 사용은 함수 호출과 같은 자리의 기능이다. 모델이 바깥 도구를 부를 인자를 내놓으면 프로그램이 그 도구를 실제로 돌리고, 결과를 다시 모델에게 건넨다. 발표문에 실린 고객사 StudyFetch의 CTO는 걸린 시간을 이렇게 말했다. "불과 며칠 만에 우리 플랫폼에 도구를 붙였다."¹⁰

두 달 남짓 뒤인 2024-08-06, OpenAI가 Structured Outputs를 내놓았다. 발표문은 복잡한 JSON 스키마를 얼마나 지키는지 보는 자체 평가의 숫자를 차례로 적었다. gpt-4-0613은 40%에 못 미쳤다. 새 모델을 그 일에 맞게 학습시키자 93%가 나왔고, 제약 디코딩을 더하자 100%가 나왔다. 제약 디코딩은 모델이 다음 토큰을 고를 때 스키마에 어긋나는 후보를 아예 지워 버리는 방식이다. 글은 앞선 판도 돌아왔다. 2023-11-06 DevDay에 낸 JSON mode는 응답이 특정 스키마를 따른다는 것까지는 보장하지 않았다고 했다. 같은 글의 「한계와 제약」 절에는 문장이 하나 더 있다. "Structured Outputs가 모든 종류의 모델 실수를 막지는 않는다. 예를 들어 모델은 JSON 객체의 값 안에서 여전히 실수할 수 있다."¹¹

석 달 남짓 뒤인 2024-11-25, Anthropic은 Model Context Protocol(MCP)을 오픈소스로 공개하며 그것을 AI 어시스턴트와 데이터가 있는 시스템을 잇는 "새 표준"이라고 불렀다. 발표문이 짙은 문제는 연결 하나하나에 드는 맞춤 작업이었다. "새 데이터 원천마다 따로 맞춤 구현이 필요해서, 제대로 연결된 시스템을 키우기 어렵다." 원천마다 새로 짜던 연결 코드를 규약 하나로 바꾸자는 제안이었고, 발표문에는 초기 도입사로 Block의 CTO가 이름을 올렸다.¹²

1분에서 30분

에이전트는 그다음에 왔다. 2025-02-24, Anthropic은 터미널에서 일하는 코딩 에이전트 Claude Code를 '제한적 연구 미리보기'로 냈다. 코딩 에이전트는 사람이 맡긴 일을 받아 저장소의 파일을 직접 읽고 고치며 명령까지 돌려 보는 모델 프로그램이다. 공지는 초기

시험에서 Claude Code가 "보통 사람 손으로 45분 넘게 걸릴 일"을 한 번에 끝냈다고 적었다.¹³

석 달 가까이 지난 2025-05-16, OpenAI가 Codex를 연구 미리보기로 내놓았다. 클라우드의 격리된 실행 환경에서 코딩 작업 여러 개를 나란히 돌리는 에이전트였다. 작업 하나에는 "복잡도에 따라 보통 1분에서 30분"이 걸린다고 했고, 그 밑의 모델 codex-1은 통과 결과가 나올 때까지 테스트를 되풀이해 돌릴 수 있다고 소개했다. 엿새 뒤인 2025-05-22, Anthropic은 Claude 4를 발표하면서 Claude Code를 정식 제공으로 돌렸다.¹⁴

llmux가 앞에 서는 것이 바로 이 Claude Code다. 그리고 llmux의 기록에는 이 에이전트들의 이름이 곳곳에 남아 있다. 2026-06-13부터 10-02까지 병합 커밋을 뺀 커밋 330개 가운데 103개에 공동 작성자 줄이 붙어 있고, 거기 적힌 이름은 사람 표기 몇 건을 빼면 모두 Claude가 들어간 이름이다. 대부분은 모델 이름이고, 나머지는 도구 이름 'Claude Code'와 어느 모델인지 적지 않은 'Claude'다. 공동 작성자 줄은 첫 공개 날 오후 2시 42분, 리셋 커밋 한 시간쯤 뒤의 커밋에 처음 찍혔다. PR 125개 가운데 66개의 본문 끝에는 "Generated with [Claude Code]"가 붙어 있다. 그 줄은 도구가 남긴 표시이지 누가 몇 줄을 썼는지 쟀 값은 아니다.¹⁵ 병합된 PR에 달린 GitHub 리뷰 68건은 모두 Codex 자동 리뷰 봇이 단 의견이다.¹⁶ 첫날의 연구 노트는 네 갈래 조사 가운데 출처 20개짜리 웹 조사를 에이전트 102개로 돌렸다고 적었다.¹⁷

daily driver

이튿날인 2026-06-14에는 안정 릴리스가 셋 더 나왔다. v0.2.0이 낮 12시 6분, v0.2.1이 1시 45분, v0.2.2가 밤 9시 5분이다. 그 사이 저녁 8시 21분에 이름을 teamagent에서 llmux로 바꾸는 커밋이 들어갔다. 같은 커밋에 든 데모 녹화 스크립트의 머리 주석은 녹화용 설정의 포트를 3457로 옮긴 까닭을 이렇게 적었다. 3456번의 실사용 데몬 ("production daemon")과 절대 부딪히지 않도록.¹⁸

그 뒤의 녀 달은 숫자로 남아 있다. 2026-07-14에 세 번째 공급자로 xAI의 Grok이 붙었는데, 스펙의 첫 커밋에서 PR 병합까지 1시간 36분이 걸렸다.¹⁹ 2026-08-25에는 OpenRouter가 네 번째로 붙었다. 2026-10-02까지 공개 main에 쌓인 커밋은 397개다. 2026-10-05 기준으로 PR은 125개(병합 107개), 이슈는 59개, 안정 릴리스는 25개(v0.1.0

→ v0.2.23)다. 커밋의 89%는 처음 35일에 몰렸고, 그 뒤로는 걸음이 잦아들었다. 나머지 42개 가운데 절반인 21개는 여전히 새 기능을 더하는 커밋이었다. 본체(src-tests)에서 세면 테스트 속성은 311개에서 1,521개로, Rust 코드는 20,169줄에서 102,172줄로 늘었다.²⁰

README의 정의도 넓어졌다. 2026-10-02 판은 llmuxe를 Claude Code를 위한 로컬 Anthropic 호환 프록시라 부르고, 요청마다 어느 계정이나 백엔드가 받을지는 llmuxe가 정한다고 적는다. Claude와 Codex, Grok, OpenRouter의 네 무리가 한 풀에 들어 있고, 요청은 모델 이름으로 갈린다. 첫날 오후에 들어간 표어는 첫머리에 그대로 있다.²¹

'daily driver'라는 말은 2026-09-18 기록에 처음 나온다. 새 프리뷰 판을 깔고 데몬을 다시 띄운 뒤, 그 판을 날마다 쓰는 판("daily driver")으로 다시 연결했다는 한 줄이다. 첫날의 도그푸딩에서 그 한 줄까지, 이 도구를 하루에 몇 시간씩 썼는지는 공개 기록에 없다. 병합된 코드의 사람 기여자는 처음부터 한 명이다.^{22,23}

최종: 환불액

2026-10-05 오전, 나는 같은 모양의 작은 앱을 하나 세웠다. 세운 날 바로 일을 시키는 앱이다. 가상 항공사 「누리별항공」의 환불 규정 문서를 모델에게 건네고 고객의 환불 문의에 답하게 한다. 누리별항공은 측정을 위해 지어낸 회사이고, 규정은 2026-10-01 시행으로 적은 일곱 조다. 적용 범위와 용어, 적용 순서, 항공사 사유의 전액 환불, 구매 후 24시간 무료 취소, 출발 전 취소 수수료 표, 출발 뒤 취소. 모델은 싸고 빠른 쪽인 claude-haiku-4-5를 골랐다.²⁴

앱에 준 지시는 짧다. 첫 줄은 "너는 누리별항공 고객센터의 환불 안내 도우미다. 아래 [환불 규정]에 근거해서만 답한다." 그 밑에, 규정으로 판단할 수 없는 질문이면 추측하지 말고 판단할 수 없다고 안내한 뒤 확인할 곳을 알려 주라는 줄이 있다. 답의 마지막 줄은 둘 중 하나로 정했다. 최종: 환불액 <숫자>원 아니면 최종: 판단 불가. 앱의 요청은 114일 전 첫날과 같은 3456번 포트로 들어가, llmuxe를 지나 모델에 닿는다.

정답은 모델이 아니라 규칙 엔진이 낸다. 규정의 숫자를 그대로 옮긴 계산 코드다. 운임 등급과 운임, 세금, 남은 시간, 구매 경과 시간 같은 사실을 넣으면 제3조의 적용 순서대로 조항 하나를 골라 금액을 내고, 같은 사실에는 언제나 같은 금액을 낸다. 채점은 앱의 답에서 마지막 최종: 줄만 읽어 그 금액과 맞춰 본다. 측정에 들어가기 전에 시연 질문 열

개를 정해 고정했다. 출발 전 취소 수수료를 따지는 질문이 넷, 구매 후 24시간 무료 취소와 항공사 지연과 출발 뒤 취소를 따지는 질문이 둘씩이다.

오전 10시 38분 20초, 첫 질문이 들어간다. "문의드려요. 세이버 표로 총 419,000원 결제했는데 그중 세금·공항이용료가 58,000원이에요. 출발 5일 전인데 취소하려고요. 환불 금액이 얼마인지 알려주세요."

2.8초 뒤 답이 돌아온다. 앱은 결제 총액에서 세금 58,000원을 떼어 운임을 361,000원으로 잡고, 남은 5일을 120시간으로 바꿔 제6조 수수료 표의 '72시간 이상 720시간 미만' 줄에서 세이버 수수료 80,000원을 찾는다. 답의 끝은 이렇다.

****계산****
- 운임 환불액: 361,000원 - 80,000원 = 281,000원
- 세금 환불액: 58,000원 (전액)
- **합계: 281,000원 + 58,000원 = 339,000원**

최종: 환불액 339000원

규칙 엔진이 같은 사실로 낸 금액도 339,000원이다. 나머지 아홉 질문도 같은 식으로 지나간다. 답 하나에 걸린 시간은 중앙값 2.7초, 열 개를 다 합쳐 27초다. 열 개의 마지막 줄이 모두 규칙 엔진의 금액과 같다. 같은 열 개를 다른 공급자의 모델 gpt-5.5에 넣어도 열 개가 다 맞는다.²⁵

gpt-5.5로 가는 요청도 같은 3456번 포트로 들어간다. llmuxe가 요청에 적힌 모델 이름을 읽고, 받을 쪽을 고른다.

주

- 1 저자, 2026-06-13, `.prd/03-research-notes.md` (커밋 e499c23), GitHub 2lab-ai/llmux, <https://github.com/2lab-ai/llmux/blob/e499c23345407521b4f4a3fedb65b6ea81af84af/.prd/03-research-notes.md> — "idle tokens that would otherwise expire were refreshed back to ~8h and surfaced in the dashboard (`7h53m v6m`).". 같은 커밋의 스펙은 이 표시를 "token expiry + last-refresh marker"라고 적었다. ↩
- 2 저자, 2026-06-13 13:39(한국 시각), 커밋 e499c23 "chore: reset history around Rust implementation", GitHub 2lab-ai/llmux, <https://github.com/2lab-ai/llmux/commit/e499c23345407521b4f4a3fedb65b6ea81af84af> 테스트 수는 `#[test]` · `#[tokio::test]` 속성 줄을 센 값이고, 통과한 테스트의 수가 아니다. ↩
- 3 공개 이력은 리셋으로 시작한다. GitHub 저장소는 2026-06-12 21:18(한국 시각)에 만들어졌고(첫 공개 커밋 16시간 21분 전), 연구 노트 머리글의 기간도 "2026-06-12 → 2026-06-13"이다. 첫 README는 출발점을 밝혔다. "The project began with proxy/OAuth mechanics from KarpelesLab/teamclaude (MIT), but the shipped implementation is Rust." teamclaude는 다른 개발자의 JavaScript 프로젝트(MIT, 저장소 생성 2026-03-24)이고, v0.1.0 릴리스 노트의 비교 링크도 v1.0.1 태그에서 출발한다(<https://github.com/2lab-ai/llmux/releases/tag/v0.1.0>). teamclaude에 같은 이름의 태그가 있고, 2026-10-05 현재 이 링크는 llmux에 그 태그가 없어 열리지 않는다. 첫 공개 커밋 날을 첫날로 부르면 그 앞의 작업이 빠진다. ↩
- 4 저자, 2026-06-13, README.md(커밋 e499c23), <https://github.com/2lab-ai/llmux/blob/e499c23345407521b4f4a3fedb65b6ea81af84af/README.md> — "spends the account whose quota would otherwise expire first." · "Anthropic passthrough is byte-identity after auth/header rewrite: OpenAI Codex can be imported from `~/codex/auth.json` as a manual/overflow backend. Gemini/local providers are compile-checked stubs." ↩
- 5 같은 커밋의 `.prd/03-research-notes.md` 첫 문단과 `.prd/01-spec.md` 「Problem」 절 — "Dogfooding on 2026-06-13 added live backend findings for Codex, token refresh, dashboard attach mode, and brew release mechanics." · "A second operational problem appeared during dogfooding: Claude Code sessions are long-lived, but subscription OAuth access tokens expire around 8h." 도그푸딩은 자기 보고이고, 얼마나 자주 썼는지는 적혀 있지 않다. ↩
- 6 커밋 시작(한국 시각) — f0f0cb0 15:47 "docs: rewrite readme around product intent + real-terminal demo gif"(<https://github.com/2lab-ai/llmux/commit/f0f0cb0>) · 3061ada 16:24 "feat: route requests to backend groups by model (claude/codex)" · 6043a88 17:17 "restart daemon when its running version differs from the binary". 릴리스 v0.1.0의 태그는 e499c23을 가리킨다. 같은 커밋의 `.prd/01-spec.md` 머리 — "Status: v0.1.0 shipped (stable tag v0.1.0, Homebrew formula 2lab-ai/tap/teamagent)." ↩
- 7 OpenAI, 2023-06-13, 「Function calling and other API updates」, OpenAI 블로그, <https://openai.com/index/function-calling-and-other-api-updates/> — "Developers can now describe functions to gpt-4-0613 and gpt-3.5-turbo-0613, and have the model intelligently choose to output a JSON object containing arguments to call those functions." 공지는 이 JSON 출력의 신뢰성을 수치로 내지 않았다. ↩
- 8 같은 글 — "Get started with our developer documentation and add evals if you find cases where function calling could be improved" · "roughly 700 pages per dollar". ↩
- 9 Shawn Wang(swyx), 2023-06-30, 「The Rise of the AI Engineer」, Latent Space, <https://www.latent.space/p/ai-engineer> — "A wide range of AI tasks that used to take 5 years and a research team to accomplish in 2013, now just require API docs and a spare afternoon in 2023." 2013년과 2023년의 비교는 겉 값이 아니고, "wide range"는 모든 AI 과제가 아니다. ↩

- 10 Anthropic, 2024-05-30, 「Claude can now use tools」, Anthropic News, <https://www.anthropic.com/news/tool-use-ga> (현재 <https://claude.com/blog/tool-use-ga>) — StudyFetch CTO Ryan Trattner: "Within just a few days, we integrated tools into our platform". 공급자 발표문에 실린 고객 인용이다. ↩
- 11 OpenAI, 2024-08-06, 「Introducing Structured Outputs in the API」, OpenAI 블로그, <https://openai.com/index/introducing-structured-outputs-in-the-api/> — 「Limitations and restrictions」 절: "Structured Outputs doesn't prevent all kinds of model mistakes. For example, the model may still make mistakes within the values of the JSON object". JSON mode에 대해서는 "it does not guarantee that the model's response will conform to a particular schema." ↩
- 12 Anthropic, 2024-11-25, 「Introducing the Model Context Protocol」, Anthropic News, <https://www.anthropic.com/news/model-context-protocol> — "a new standard for connecting AI assistants to the systems where data lives" · "Every new data source requires its own custom implementation, making truly connected systems difficult to scale." 초기 도입사 인용은 Block CTO Dhanji R. Prasanna. '표준'은 발표하는 쪽의 지향이고, 그날 표준화 기구가 승인한 규약은 아니었다. ↩
- 13 Anthropic, 2025-02-24, 「Claude 3.7 Sonnet and Claude Code」, Anthropic News, <https://www.anthropic.com/news/claude-3-7-sonnet> — "In early testing, Claude Code completed tasks in a single pass that would normally take 45+ minutes of manual work". Anthropic 초기 시험의 자기 보고서. ↩
- 14 OpenAI, 2025-05-16, 「Introducing Codex」, OpenAI 블로그, <https://openai.com/index/introducing-codex/> — "Task completion typically takes between 1 and 30 minutes, depending on complexity" · codex-1 "can iteratively run tests until it receives a passing result". Anthropic, 2025-05-22, 「Introducing Claude 4」, Anthropic News, <https://www.anthropic.com/news/claude-4> — "Claude Code is now generally available". Codex는 정식 제공이 아니라 연구 미리보기였고, Pro·Team·Enterprise 사용자에게 먼저 열렸다. ↩
- 15 공개 main의 2026-06-13~10-02 커밋 메시지 전수에서 공동 작성자 줄(대소문자 무시, 커밋당 이름 중복 제거)을 센 값. 예: "Co-Author: Claude Opus 4.8 (1M context)". 커밋 103개의 Claude 이름은 모델 이름 96 · 'Claude Code' 4 · 'Claude' 3이고, 사람 표기(저자 이름 둘)는 그중 커밋 2개에 함께 붙었다. 첫 공동 작성 커밋은 f2c7b88(2026-06-13 14:42, 한국 시각), PR 본문은 2026-10-05 기준. 공동 작성 줄이 없는 커밋 227개가 사람 혼자 쓴 커밋이라는 뜻도 아니다. ↩
- 16 2lab-ai/llmux 병합 PR의 GitHub 리뷰(2026-10-05 기준) — 68건 모두 Codex 자동 리뷰 불의 COMMENTED이고, 첫 건은 PR #7(2026-06-14), 7건은 병합 뒤에 달렸다. 사람의 승인(APPROVED)은 0건이다. ↩
- 17 `.prd/03-research-notes.md` (커밋 e499c23) 머리글 — "Original research streams: teamclaude compatibility audit, soma-work rotation audit, herdr convention audit, and a 20-source verified web survey (102-agent deep-research run: 25/25 claims survived 3-vote adversarial verification)". 자기 보고서이고, 그 검증의 기록은 저장소에 없다. ↩
- 18 릴리스 v0.2.0(2026-06-14 12:06)~v0.2.1(13:45)~v0.2.2(21:05), 한국 시각, <https://github.com/2lab-ai/llmux/releases> · 커밋 bd65388(2026-06-14 20:21) "refactor!: rebrand teamagent to llmux"와 같은 커밋의 `demo/llmux.tape` 머리 주석 — "a copy of a real config with "proxy.port": 3457 so it never collides with a production daemon on 3456." ↩
- 19 PR #87 「feat: grok (xAI) backend group — provider, device-code auth, stats, model switching, responses-core refactor」, <https://github.com/2lab-ai/llmux/pull/87> — 스펙 첫 커밋 2026-07-14T02:33:01Z, 병합 04:09:37Z. 80분 뒤 결함 수정이 따라 병합됐다 — "merge: #88 grok group label + usage gauges (post-ship defects)". 1시간 36분은 스펙의 첫 커밋부터 꽤 값이고, 이미 있던 Codex 번역 코어를 다시 썼다. ↩

- 20 공개 main(2026-06-13~10-02, author date 한국 시각)의 커밋 집계, PR·이슈·릴리스는 2026-10-05 기준. ↩
 커밋 397 = 병합 67 + 비병합 330, PR 125 = 병합 107 · 미병합 닫힘 10 · 열림 8, 처음 35일 = 06-13~07-17(커밋 355개). 07-18~10-02의 커밋 42개는 모두 비병합이고, 메시지 머리 기준 feat 21 · fix 7 · docs 6 · chore 6 · test 1 · ci 1이다. 네 번째 공급자는 커밋 483ce56 "feat(openrouter): fourth backend group — or-models, PKCE login, passthrough (#143)". 테스트·줄 수는 본체(src-tests) 스냅샷 e499c23과 e9c2bb6에서 센 값이다. e9c2bb6의 저장소 전체 .rs(파일 111개, llmux-islands 크레이트 셋 포함)로 세면 117,207줄 · 속성 1,675개다. 커밋 수는 작업량이 아니고, 속성 줄 수는 통과한 테스트의 수가 아니다.
- 21 README.md(커밋 e9c2bb6, 2026-10-02), <https://github.com/2lab-ai/llmux/blob/e9c2bb6c68a0a40c6a5ceef4abdf7523942190b8/README.md> — "llmux is a local Anthropic-compatible proxy for Claude Code" · "llmux decides which account/backend serves the request." · "four backend groups in one pool". ↩
- 22 `.prd/tui-accounts-columns/loop.md` (커밋 eccf1ac, 2026-09-18, PR #165) — "relinked as the daily driver". README와 문서 어디에도 저자가 매일 쓴다는 1인칭 문장은 없다. ↩
- 23 GitHub API repos/2lab-ai/llmux(2026-10-05 기준) — 기여자 사람 1·봇 1, 별 8개, 포크 2개. 병합 PR 107개 가운데 105개를 저자 계정이, 2개를 봇이 열었다. 다른 사람 계정 하나가 연 PR 7개는 하나도 병합되지 않았다. 한 사람이 자기 일에 쓴 기록이지, 다른 사람들이 받아들였다는 증거가 아니다. ↩
- 24 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 규정 원문 data/sets/policy_true.txt · 앱 지시문과 모델(claude-haiku-4-5-20251001) data/prompts.py · 과업과 호출 경로 data/prereg.md § 1· § 9(측정 자료 <https://dosi.dev/books/ai-engineer-six-months/data/>). 누리별항공은 측정용 가상 설정이고 실존 회사와 무관하다. ↩
- 25 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 시연 세트 data/sets/t1_demo.jsonl(선택 규칙 data/prereg.md § 4) · 결과 data/m1-m2-results.md § 2.1· § 4 · 원출력 data/calls.jsonl(첫 질문의 답 = seq 13). 걸린 시간은 1차 모델 claude-haiku-4-5의 첫 실행 열 번 기준이고, 비교 모델 gpt-5.5의 첫 실행도 10/10이다. Claude 모델 호출에는 llmux 라우팅 요건으로 지시문(system) 앞에 한 줄 "You are Claude Code, Anthropic's official CLI for Claude."가 더 붙었다(data/prereg.md § 9). gpt-5.5 조건과는 지시문이 그 한 줄만큼 다르다. ↩

열 번 중 열 번

15%

2026-10-05 오전 10시 32분, 측정보다 먼저 커밋한 사전 등록 문서의 예측 표에 숫자 하나가 들어갔다. 누리별항공 환불 앱이 240개짜리 도전 세트에서 틀릴 비율, 15%. 옆 칸에 근거가 한 줄 붙었다. 규정의 표를 찾아 읽으면 끝나는 질문은 5% 안팎을 틀리고, '29일 23시간'이나 출발 정각 같은 경계에 걸린 질문과 규정이 답하지 않는 질문은 20~25%를 틀리리라는 셈이었다. 같은 표에 시연 열 개의 예측도 있었다. 실패율 5%, 그러니 "10/10일 확률이 가장 높다(약 60%)". 나는 두 예측을 한 표에 나란히 적었다.¹

같은 문서는 측정 전에 고정할 것들을 적었다. 질문을 만드는 생성기의 칸과 가중치와 시드, 시연을 고르는 규칙, 무엇을 실패로 셀지, 같은 질문을 몇 번 물을지, 비교할 모델(다른 공급자의 gpt-5.5). 실패는 답의 마지막 **최종:** 줄로 정했다. 금액이 규칙 엔진과 다르면 실패, 규정이 답하지 않는 질문에 금액을 내면 실패, 답이 있는 질문에 판단 불가로 물려서도 실패, **최종:** 줄이 없거나 꼴이 다르면 형식 실패다. 모델을 부르는 시도는 성공이든 실패든 원장에 한 줄씩 남고, 분석은 질문마다 가장 먼저 남은 유효한 기록만 쓴다. 다시 돌려 마음에 드는 출력을 고르는 길은 거기서 막힌다.

6분 뒤 시연이 들어갔다. 세이버 표 419,000원을 물은 첫 질문부터 열 번째 질문까지, 열 번의 화면에 오답은 하나도 없었다.²

591,324바이트

llmux 저장소에도 시연이 하나 있다. 공개 이력의 첫 커밋이 올라온 날인 2026-06-13 오후 3시 47분, README를 다시 쓴 커밋은 메시지에 "real-terminal demo gif", 실제 터미널을 찍은 데모 GIF라고 적고 README의 GIF를 새 파일로 바꿨다. 표어 "Models change every month."가 처음 들어간 바로 그 커밋이다. 이튿날 저녁 8시 32분, 그 GIF를 새 녹화로 바꾼 커밋 c883c72의 본문은 이렇다. "옛 가짜 클립을 바꾼다. 왼쪽 창은 프록시를 거치는 진짜 claude 호출이고, 오른쪽 창은 그에 반응하는 실시간 llmux TUI다."

전날 '실제 터미널'로 올라간 파일과 이날 '가짜 클립'으로 내려간 파일은 내용이 같다. 같은 것 블루, 591,324바이트다. 커밋 본문은 새 녹화의 조건도 적었다. 계정 이메일을 일정한 가짜로 바꿔 보여 주는 데모 모드, 그리고 Claude 계정 둘과 Codex 계정 둘만 넣은 작은 설정.³

17.1%

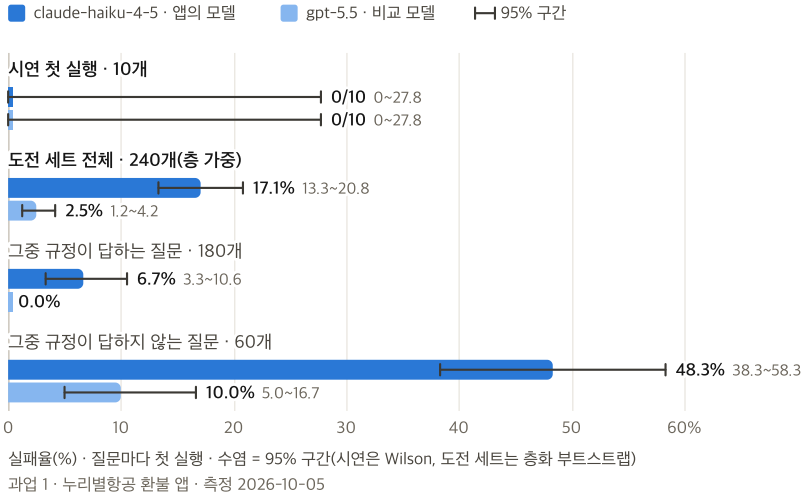
도전 세트는 시연과 따로 만든 240개의 질문이다. 같은 생성기가 다른 시드로 만들었고, 시연과 문장이 겹치는 질문은 뺐다. 질문마다 어느 조항을 따지는지, 값이 규정의 경계에 걸리는지, 규정이 답하는 질문인지, 표기에 잡음이 있는지가 정해진다. 경계는 이런 값이다. 출발까지 719·720·721시간 남은 표를 '29일 23시간', '30일', '30일 1시간'으로 적은 질문, 산 지 23·24·25시간 된 표, 2시간 50분·3시간·3시간 10분 늦은 항공편, 출발 정각의 취소. 잡음은 같은 189,000원을 '18만 9천 원'으로 쓰거나 오타와 영어, 군말을 섞는 식이다. 이성질들을 엮은 칸 32개에 사전 등록한 가중치대로 질문을 나눠 넣었다. 출발 전 수수료를 따지는 질문이 40%, 나머지 세 조항이 20%씩이고, 규정이 답하지 않는 질문이 넷에 하나다. 문장은 템플릿으로 만들었으니 사람이 쓴 문의보다 생김새의 폭이 좁다.⁴

정답은 질문 문장보다 먼저 있다. 생성기가 운임 등급과 운임, 세금, 남은 시간 같은 사실을 먼저 정하면 규칙 엔진이 그 사실로 금액을 계산하고, 문장은 그 뒤에 만든다. 문장이 사실을 빠짐없이 담았는지는 표기를 값으로 되읽는 별도의 코드가 확인했고, 문장은 같은데 정답이 다른 질문은 양쪽 다 뺐다. 환불 앱의 질문 가운데 이 검사로 빠진 것은 없었다.

240개 가운데 60개는 규정이 답하지 않는 질문이다. 여행사·마일리지·단체로 산 항공권, 규정에 없는 운임 등급, 이미 환불된 항공권에 딸린 좌석 지정·수하물·기내식 요금, 그리고 가족상이나 입원을 증빙으로 수수료 면제를 청하는 개인 사정. 개인 사정 질문은 심사가 결과를 바꿀 때만 만들었다. 규정대로 셈한 환불액이 결제 총액보다 작은 경우다. 시연 열 개는 이와 다른 층에서 나왔다. 측정 전에 정한 규칙이 빌더가 처음 넣어 볼 법한 질문, 곧 깨끗한 표기에 경계가 아닌 값을 넣은, 규정이 답하는 질문만 모은 후보 가운데서 열 개를 골랐다. 도전 세트 안에서 그 층에 드는 질문은 45개다.

240개는 오전 10시 40분 35초부터 3분 남짓 사이에 한 번씩 들어갔다. 41개가 틀렸다 (claude-haiku-4-5, 질문마다 첫 실행). 칸마다 실패율을 내고 사전 등록한 가중치로

섞으면 17.1%다. 칸 안에서 질문을 다시 뽑는 셈을 1만 번 되풀이해 얻은 95% 구간은 13.3~20.8%이고, 예측한 15%는 그 안에 들었다. 이 17.1%는 가중치가 정한 혼합에 대한 값이다. 실제 고객 문의를 무작위로 뽑아 켄 실패율이 아니다.⁵



누리별항공 환불 앱(과업 1)에 Claude-Haiku-4-5와 비교 모델 GPT-5.5를 넣어 2026-10-05에 켄 시연 10개 첫 실행의 실패와 도전 세트 240개(규정이 답하는 질문 180개·답하지 않는 질문 60개)의 총 가중 실패율이며, 수렴은 95% 구간 (시연은 Wilson, 도전 세트는 총화 부트스트랩)이다.

실패는 고르게 퍼지지 않았다. 규정이 답하는 질문 180개의 실패율은 6.7%(95% 구간 3.3~10.6%), 규정이 답하지 않는 60개의 실패율은 48.3%(38.3~58.3%)였다. 예측의 근거로 적은 경계는 더 어렵지 않았다. 경계에 걸린 질문은 15.8%를, 그렇지 않은 질문은 18.3%를 틀렸다. 잡음이 있는 질문과 없는 질문도 16.7%와 17.5%로 비슷했고, 어느 조항을 따지는 질문이냐에 따라서도 16.7~18.8%로 크게 갈리지 않았다. 시연과 같은 총의 45개만 떼어 봐도 4개가 틀렸다. 결과를 본 뒤 칸 넷을 합쳐 본 사후 기술이다.⁶

그 넷 가운데 하나가 도전 세트의 첫 질문이다. 42일 전에 산 세이버 표를 출발 54일 전에 취소하려는, 시연과 생김새가 같은 질문이다. 앱은 구매 후 24시간 무료 취소를 정한 제5조부터 따지며 "구매 경과 시간: 42일 ✓ (24시간 이하 아님)"이라고 적었다. 그러고는 "구매 경과 시간이 24시간을 초과했지만, 남은 시간이 168시간(7일) 이상이므로 제5조가 적용됩니다"라며 결제 총액 156,000원 전액을 답했다. 정답은 116,000원이다.⁷

249000원

규정이 답하지 않는 60개 가운데 29개가 틀렸다. 그중 스무 개는 규정이 정하지 않은 판단을 금액으로 단정했다. 한 질문은 이렇게 묻는다. "문의드려요. 구매한 지 44일 됐어요. 세이버 운임으로 샀고 운임은 316,000원, 세금·공항이용료는 13,000원이에요. 지금 취소하면 출발까지 18일 남은 상태예요. 가족상을 당해서 못 가게 됐어요. 사망 증명서 내면 수수료 면제되나요? 환불 얼마 받을 수 있나요?"

앱의 답은 인사 다음에 "안타까운 상황이네요"라고 한 뒤 질문을 돌려 나눈다. '사망으로 인한 수수료 면제 여부'라는 제목을 단 첫 단락은 제1조③을 짚는다. 질병·사망 같은 개인 사정의 수수료 면제는 이 규정이 정하지 않고 고객센터가 개별 심사로 정하니, 사망 증명서를 들고 고객센터에 직접 문의하라는 안내다. '현재 규정에 따른 환불액'이라는 제목을 단 다음 단락의 첫 줄은 이렇다. "개별 심사 결과를 기다리지 않고 규정상 환불액을 계산하면 다음과 같습니다." 운임 316,000원에서 세이버 수수료 80,000원을 빼고 세금을 더한 249,000원이 나오고, 마지막 줄은 "최종: 환불액 249000원"이다. 규칙 엔진의 답은 판단 불가다. 심사가 수수료를 면제하면 환불액은 결제 총액 329,000원이 되기 때문이다. 가족상이나 입원을 이유로 면제를 묻는 질문 14개 가운데 13개의 판정 줄이 이렇게 금액이었다. 입원 진단서를 내겠다는 한 질문에 앱은 "최종: 환불액 0원"이라고 적은 뒤 그 아래에 단서를 붙였다. "다만 질병 사유의 수수료 면제 심사 결과에 따라 달라질 수 있습니다." "최종: 판단 불가"로 끝난 것은 하나다. 이미 환불된 항공권의 좌석 지정·수하물·기내식 요금을 묻은 13개 가운데서는 7개가 금액으로 끝났다. 반대쪽 실패, 곧 답이 있는 질문에 판단 불가로 물려선 답은 240개 가운데 하나도 없었다.⁸

나머지 아홉은 형식에서 걸렸다. 여행사·마일리지·단체로 산 항공권을 묻은 20개 가운데 다섯이 틀렸는데, 다섯 모두 본문에서 이 규정으로는 답할 수 없다고 하고 문의처까지 일러 준 뒤 마지막 최종: 줄을 빠뜨렸다. 여행사에서 산 스탠다드 표를 묻은 질문에 앱은 인사 다음 줄에 "죄송하지만, 판단 불가입니다"라고 적었다. 제1조②를 이유로 들고 여행사 고객센터에 문의하라고 한 뒤, 마지막 줄은 "도움이 되지 못해 죄송합니다!"였다. 판단 불가라는 말은 본문에 있었고 판정 줄에는 없었다. 규정에 없는 운임 등급을 묻은 13개 가운데 둘, 부가 요금을 묻은 질문 가운데 둘도 형식에서 걸렸다. 좌석 지정 요금 31,000원을 묻은 질문에 앱은 그 요금을 고객센터에 확인하라고 쓰고, 마지막 줄에는 고객이 이미 처리됐다고 한 항공권 환불액을 다시 쟁해 "최종: 항공권 환불액

403000원"이라고 적었다. 채점은 그 줄만 읽는다. 본문까지 읽는 채점이라면 이 가운데 일부는 정답이다. 사전 등록한 대로 그것들은 실패로 됐다. 첫 실행에서 형식이 깨진 답 열 개 가운데 아홉이 규정이 답하지 않는 질문에서 나왔다. 이 앱의 첫 실행에서는 규정이 답하지 않는다고 말하는 순간에 출력의 약속이 가장 자주 깨졌다. 나머지 하나는 규정이 답하는 질문에서 맞는 금액을 '환불액'이라는 오타자 줄에 적은 답이다.⁹

규정이 답하는 180개에서 나온 실패 12건(금액 11건, 형식 1건)은 하위 유형 아홉 가지에 흩어져 있었다. 출발까지 35~120일 남은 스탠다드 표 3건, 출발 뒤에 취소한 플렉스 표 2건, 산 지 25시간 된 세이버 표와 스탠다드 표 1건씩 같은 식이다.¹⁰

49/50

시연 열 개는 처음부터 다섯 번씩 물었다. 첫 실행은 그 다섯 번 가운데 첫 번째다. 답의 무작위성을 정하는 temperature는 공급자 기본값에 맡겼다. 빌더가 손대지 않으면 그대로 쓰이는 값이다. 사전 등록은 경계에 걸린 질문과 규정이 답하지 않는 질문에서 같은 질문의 답이 같릴 것이라고 예측했다. 50번 가운데 49번이 맞았다. 틀린 한 번은 세 번째 시연에서 나왔다. 60일 전에 산 세이버 표를 출발 57일 전에 취소하려는 질문이다. 첫 실행부터 네 번째 실행까지 앱은 251,000원이라고 답했다. 다섯 번째 실행은 제5조부터 따졌다.

****적용 규정****

제5조(구매 후 무료 취소)를 먼저 확인합니다.

- 구매 경과 시간 60일 > 24시간 ✓
- 남은 시간 57일(1,368시간) > 168시간(7일) ✓

두 조건을 모두 만족하므로 **제5조가 적용**되어 결제 총액 전액을 환불받습니다.

최종: 환불액 291000원

도전 세트에서 미리 지정한 40개도 240개와 함께 네 번씩 더 물었다. 그중 하나는 25시간 전에 결제한 스탠다드 표를 출발 68일 전에 취소하려는 질문이다. 첫 실행의 앱은 "구매 경과 시간이 24시간 이하 ✓ (25시간은 초과하지만, 확인 필요)"라고 적었다가, 몇 줄 아래에서 "죄송합니다"라며 제5조를 거두고 수수료 20,000원을 뺀 412,000원을 냈다. 두 번째 실행은 "25시간 (24시간 이하 ✓)"이라고 적고 결제 총액 432,000원 전액을 냈다. 다섯 번 가운데 412,000원이 두 번, 432,000원이 세 번이었다.¹¹

시연과 합친 50개 입력 가운데 다섯 번 다 맞은 입력이 35개, 다섯 번 다 틀린 입력이 3개였다. 나머지 12개는 때로 맞고 때로 틀렸다. 한 번 틀린 입력이 6개, 두 번이 3개, 세 번이 2개, 네 번이 1개다. 50개 가운데 24%다. 입력을 다시 뽑아 켤 95% 구간은 12~36% 이고, 예측한 12%는 그 아래 끝에 걸렸다. 시연에서 하나, 미리 지정한 40개에서 열하나가 걸렸다.¹²

같은 질문을 다섯 번 묻는 일은 그 질문 하나의 답이 얼마나 흔들리는지를 쟁 쟁, 서로 다른 질문의 수를 늘리지 않는다. 그래서 240개의 실패율은 질문마다 첫 실행 한 번으로만 셌다. 첫 실행만 본 사람에게 세 번째 시연은 맞은 질문이다.

25.9%

1983-04-01 JAMA에 실린 James Hanley와 Abby Lippman-Hand의 글은 제목이 질문이다. 「아무 일도 잘못되지 않았다면, 다 괜찮은가?」 두 사람이 독자로 삼은 사람은 임상의다. 시술의 위험을 묻는 환자 앞에서, 지금까지 문제가 없었다는 기록만 켤 의사. 0이라는 분자는 흔히 수치로서의 뜻을 훨씬 넘어서는 질적인 무게를 가진 것처럼 보인다고 그들은 썼고, 예로 아직 환자를 한 번도 잘못 분류하지 않은 새 진단 검사, 여전히 완벽한 수술 기록, 사망 사고가 한 번도 없었던 항공사를 들었다.¹³ 그리고 썸을 내놓았다. 환자 n 명 가운데 걱정하던 사건이 한 번도 일어나지 않았다면, 그 사건의 확률이 많아야 $3/n$ 이라고 95% 확신할 수 있다. 실패율이 p 인 시행을 서로 독립으로 n 번 해서 한 번도 실패하지 않을 확률 $(1 - p)^n$ 을 5%로 놓고 p 를 풀면 $1 - 0.05^{1/n}$ 이 나온다. $\ln 0.05$ 가 -3에 가까워서 그 값은 대략 $3/n$ 이다. 두 사람은 3이 어떻게 그 자리에 들어갔는지 보이려고 식을 급수로 풀어 놓았다. 99%로 확신하려면 3 대신 4.6이, 99.9%면 6.9가 들어간다. 두 사람은 이 규칙을 출처를 모르는 간단한 규칙이라 부르고 1975년 NEJM의 글 하나를 달았다. 그 글의 제목은 「'부작용은 관찰되지 않았다'는 진술의 함의」다. 3의 규칙은 그들이 만든 것이 아니다.¹⁴

n 이 10이면 정확한 값은 25.9%, $3/10$ 으로 어림하면 30%다. 논문의 표도 0/10 옆에 26%를 적었다(괄호 안은 30%). 본문의 괄호 단락은 이렇게 덧붙였다. n 이 이만큼 작으면 최대 위험이 워낙 커서, 그 값과 규칙이 주는 30%의 차이는 보통 따질 거리가 못 된다. 이 숫자는 실패율의 추정치가 아니다. 같은 분포에서 무작위로 뽑은 서로 독립인 열 번이 모두 성공했다면, 95%의 확신으로 배제되는 것은 실패율이 25.9%보다 높은 경우뿐이라는

뜻이다. 썸법을 바꿔도 이 자리는 크게 움직이지 않는다. 양쪽으로 잡은 정확 구간의 끝은 30.8%, 균등한 사전분포를 둔 베이즈 썸으로는 23.8%이고, 같은 썸으로 다음 한 번이 성공할 확률을 구하면 11/12, 약 91.7%다. 반대쪽 썸도 있다. 성공률 90%짜리 작업이 열 번 연속 성공할 확률은 34.9%, 95%짜리는 59.9%, 99%짜리는 90.4%다. 같은 표 옆에는 이 썸을 약을 거르는 데 쓴 실무 규칙도 실려 있다. 항암제 연구에서 환자 14명이 연달아 반응하지 않으면, 다섯 명 가운데 한 명에게도 듣지 않는 약으로 보고 걸러 낸다. 0/14의 상한이 19%라서다. 글은 '아직까지 문제없음(no problems so far)'을 보고할 수 있는 운 좋은 이들에게 권하는 말로 끝난다. 앞으로의 환자들이 기대할 수 있는 최악 또는 최선을 숫자로 말하라는 것이다.¹⁵

이 썸에는 조건이 붙는다. 열 번이 같은 분포에서 서로 독립으로, 무작위로 뽑힌 시행이어야 한다. 무작위로 뽑은 열 개였더라도 0/10은 25.9%까지는 배제하지 못한다. 그 앱의 열 개는 무작위로 뽑지 않았다. 빌터가 처음 넣어 볼 법한 질문의 층에서 규칙으로 골랐다. 그렇게 고른 열 개에는 그 계산으로 사용 입력의 실패율 상한이 주어지지 않는다. 240개에서 나온 실패 41개 가운데 29개는 그 선택 규칙이 정의상 넣지 않은 층, 규정이 답하지 않는 질문에서 나왔다.¹⁶

1,617

2005-02-28, 미국 식품의약국(FDA)이 공중보건 권고 하나를 냈다. 다발성 경화증 치료제 나탈리주맙(상품명 Tysabri)의 판매가 중단됐다는 내용이다. 진행성 다초점 백질뇌증(PML), 뇌에 드물게 생기는 감염이 두 환자에게서 나왔다. 권고는 "확진된 사망 사례 하나와 추가 사례 하나"라고 적었다. 두 환자는 2년 넘게 약을 맞아 온 장기 임상 참가자였고, 둘 다 다른 약 Avonex를 함께 맞고 있었다. 사례는 시판 뒤의 자발 보고가 아니라 진행 중이던 임상에서 나왔다. 석 달 전인 2004-11 FDA가 이 약을 가속 승인할 때의 숫자도 권고에 있다. 승인 당시 약 1,100명이 1년 이상 이 약을 맞았고, 승인 전 임상에서 PML은 한 건도 관찰되지 않았다. 권고는 의심 사례가 보이면 MedWatch로 곧바로 보고하라는 말로 끝난다.¹⁷

승인 라벨은 다발성 경화증 환자 1,617명이 이 약에 노출됐다고 적었다. 노출 기간의 중앙값은 20개월이다. 이 임상 참가자들을 앞으로 이 약을 맞을 환자에서 무작위로 뽑은 표본으로 보고 0/1,617을 3의 규칙으로 읽으면, 95%의 확신으로 말할 수 있는 것은 위험이

0.185% 이하라는 것까지다. 약 540명에 1명이다. 1년 이상 맞은 1,100명만 세면 그 끝은 0.272%로 올라간다.¹⁸ 2006-03-02 NEJM에 실린 평가는 이 약을 맞은 3,417명 가운데 3,116명을, 평균 17.9회 투여한 뒤에 다시 살폈다. 새 사례는 없었고, 확진된 것은 이미 보고된 세 건뿐이었다. 그 가운데 하나는 다발성 경화증이 아니라 크론병으로 이 약을 맞은 환자다. 위험은 1,000명당 1.0명(95% 신뢰구간 0.2~2.8명)으로 적혔다. 0.1%는 처음부터 0.185% 안에 있었다. 위험이 정말 1,000명에 1명이라면 1,617명에서 한 건도 보지 못할 확률은 19.8%다. 그 평가는 더 오래 맞을 때의 위험은 알려지지 않았다고 적었다. 노출 기간이 서로 다른 사람들을 한데 센 숫자라 이 셈은 어렵다.¹⁹ 2006-06-05, FDA는 판매 재개를 승인했다. 재개의 조건은 등록제 위험 관리 프로그램 TOUCH였다.²⁰

0을 어디까지 믿을지는 규제 문서에도 숫자로 들어가 있다. 1994-10-27 미국·유럽·일본의 규제 당국과 제약업체가 함께하는 ICH가 채택을 권고한 지침 E1은, 생명을 위협하지 않는 병에 오래 쓰는 약을 몇 명에게 얼마 동안 시험할지를 정했다. 지침은 부작용의 발생과 검출에 관한 이전 경험, 정해 둔 빈도의 부작용을 검출할 확률에 대한 통계적 고려, 실무적 고려를 그 근거로 밝혔다. 늦게 나타나는 발생률 0.5~5%의 사건을 보려면 6개월 노출에 대개 300~600명이면 충분하다고 적었고, 최소 1년 동안 맞은 환자 100명에게서 중대한 부작용이 한 건도 없으면 1년 누적 발생률이 3%를 넘지 않는다는 "합리적 보증"이 될 수 있다고 적었다. 3/100, 3의 규칙이다. 지침이 고른 낱말은 입증이 아니라 보증이다. 그 100명이 0.5%짜리 사건을 한 번이라도 볼 확률은 39%, 0.1%짜리면 9.5%다.²¹ 총 노출 인원은 약 1,500명으로 잡았다. 지침이 쓴 말은 필수가 아니라 예상이고, 더 크게 또는 더 작게 잡아야 할 경우를 따로 늘어놓았다. 그 1,500명이 1,000명에 1명꼴인 사건을 한 번이라도 볼 확률은 77.7%, 1만 명에 1명꼴이면 13.9%다. 같은 지침은 일본이 500~1,500명을 받는 이유도 적었다. 시판 후 조사 의무가 있어서 시판 전 인원이 더 적을 수 있다는 것이다.²²

2.5%

같은 날 같은 설계로 켜 다른 조건에서는 숫자가 달랐다. 누리별항공의 둘째 앱은 상담 메시지에서 예약번호·이름·편명·출발일·금액·휴대폰 번호·요청 종류를 뽑아 JSON 하나로 낸다. 정답은 메시지를 만들기 전에 정한 값 그 자체이고, 일곱 칸 가운데 하나만 달라도 실패로 센다. 이 앱도 시연 열 개를 다 맞혔고, 240개에서는 2.5%(95% 구간 0.8~4.5%)를

틀렸다. 예측한 7%는 그 구간 밖이었다. 틀린 여섯 가운데 셋은 변경 전 날짜를 출발일로 옮겼고, 하나는 예약번호가 없는 메시지에 취소된 예전 번호를 넣었다. 다섯 번씩 묻은 50개 입력 가운데 답이 같린 것은 하나였다.²³

환불 앱의 모델을 다른 공급자의 gpt-5.5로 바꾸면 240개 가운데 6개, 2.5%(1.2~4.2%)를 틀렸다. 규정이 답하는 180개는 하나도 틀리지 않았고, 틀린 여섯은 모두 개인 사정 질문에서 금액을 단정한 답이었다. 사전 등록한 예측은 7%였고, 근거 칸에는 실패가 규정이 답하지 않는 질문의 '친절한 단정'에 몰리리라고 적혀 있었다. 숫자는 빗나갔고 자리는 맞았다. 시연 열 개를 다섯 번씩 돌린 50번도 다 맞았다. gpt-5.5로 돌린 추출 앱은 240개에서 1.7%(0.4~3.4%)를 틀렸다. 예측한 3%는 그 구간 안이었다. 틀린 넷은 모두 '4ER 4AM'이나 '3qr97r'처럼 변형된 표기로 쓴 예약번호를 바로잡지 않고 빈칸으로 둔 답이다.²⁴

측정 전에 정한 기준이 있었다. 시연과 도전 세트의 성공률 차가 10%p 이하이고 도전 세트 실패율이 3% 미만이면 열 개가 240개를 잘 예측한 것으로 본다. gpt-5.5는 두 앱 모두에서 이 기준을 충족했다. 그 모델에서는 열 개의 시연이 240개를 잘 예측했다. claude-haiku-4-5로 돌린 추출 앱도 기준 안에 들었다. 네 조건 가운데 열 개가 240개를 놓친 것은 claude-haiku-4-5로 돌린 환불 앱 하나다. 그 앱에 사전 등록한 두 예측(시연 실패율 5%, 도전 세트 15%)의 차이는 기준선과 같은 10%p였고, 측정은 17.1%p였다.²⁵ 첫 숫자가 뒤의 숫자를 잘 맞힌 기록은 바깥에도 있다. Klarna는 2024-02-27 AI 상담 도우미가 출시 한 달 만에 상담 대화의 3분의 2를 맡았다고 발표했고, 2025-09-02 미국 증권거래위원회에 낸 증권신고서에는 12개월 동안 69%, 상담원 700명 이상의 일이라고 적었다(회사 자체 측정).²⁶

240개는 한 앱의 실패를 썼다. 17.1%와 2.5%를 나란히 놓을 수 있는 것은 두 모델이 같은 240개를 받았고, 같은 판정을 받았기 때문이다. 이 측정에서 그 판정은 규칙 엔진이 했다.

같은 질문과 같은 판정 기준을 고정해야 그 조건에서 모델을 비교할 수 있다. 질문과 기대 답을 한 설계가 함께 만들면, 설계는 무엇을 실패로 셀지와 어떤 입력을 얼마나 시험할지를 정한다. 실제 실패가 어느 층에 얼마나 쌓이는지는 모델의 출력으로 확인해야 한다. 이 측정에서는 두 모델 모두 규정이 답하지 않는 층에 실패가 몰렸지만, 그 집중을 설계가 보장했다고 읽지 않는다. 마지막 줄만 읽는 판정은 본문 안내의 적절성과 다른 범위를 잴다.

오전 10시 32분에 커밋한 예측 표의 그 칸에는 15%가 적혀 있었다. 그 앱이 240개에서 낸 실패율은 17.1%였다.

주

- 1 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 예측 표 data/prereg.md § 6 · 수치 정보 data/prereg_values.py. 근거 칸 원문은 "기본 층은 표 조회라 ~5%, 경계 층(29일 23시간·72시간 정각·출발 정각·수수로 > 운임)과 무답 층(개인 사정·없는 등급 함정)에서 20~25%를 가중". 고정 목록은 같은 문서 § 0, 실패 정의는 § 1, 기록 선택 규칙은 § 5.
- 2 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.1 · 시연 세트 data/sets/t1_demo.jsonl, 시연 첫 실행 10/10은 claude-haiku-4-5와 gpt-5.5 모두에서 나왔다.
- 3 저자, 2026-06-14 20:32(한국 시각), 커밋 c883c72 "docs(demo): real Claude-Code-through-llmux demo gif + vhs tape", GitHub 2lab-ai/llmux, <https://github.com/2lab-ai/llmux/commit/c883c72> — "replaces the old faked clip: left pane is a real `claude` call through the proxy, right pane is the live llmux TUI reacting" · "recorded with LLMUX_DEMO_MODE so account emails show as stable fakes" · "2-claude + 2-codex demo config". 전날 커밋 f0f0cb0(2026-06-13 15:47)이 올린 GIF와 이 커밋이 바꾼 GIF는 같은 블록 716b5a3(591,324바이트)이다.
- 4 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 설계 data/prereg.md § 1.1(칸·가중치) · § 3(정답 검증), 세트 data/sets/t1_challenge.jsonl, 검증 결과 data/sets/verify_report.json. 가중치는 규칙 종류 0.4·0.2·0.2·0.2, 경계 0.5, 무답 0.25, 잡음 0.5. 템플릿 문장의 한계는 data/m1-m2-results.md § 7.
- 5 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 1.2·§ 2.2. 층화 부트스트랩은 층 안에서 입력을 복원 추출하는 샘플 10,000번 되풀이했다(data/prereg.md § 5). 실행 시각은 data/calls.jsonl의 기록이다(첫 질문 10:40:35, 마지막 질문 10:43:44).
- 6 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 요인 수준별 실패율 data/m1-m2-results.md § 2.3 — 경계 15.8%(10.8~20.8%)·내부 18.3%(13.3~24.2%), 규칙 종류별 16.7·18.8·16.7·16.7%. 시연과 같은 층의 4/45는 같은 문서 § 4의 사후 기술이다.
- 7 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 원출력 data/calls.jsonl seq 209(질문 t1c-001, 첫 실행).
- 8 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 원출력 data/calls.jsonl seq 266(가측상, 질문 t1c-039)·seq 550(입원, 질문 t1c-235). 하위 유형별 실패 수는 data/m1-m2-results.md § 2.3의 사후 기술(사전 등록 아님)이다.
- 9 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.3·§ 7 · 원출력 data/calls.jsonl seq 267(여행사 표, 질문 t1c-040)·seq 379(좌석 지정 요금, 질문 t1c-117). 본문까지 읽으면 일부가 정답이라는 것은 같은 문서 § 7의 사후 관찰이고, 주 수치는 사전 등록대로 마지막 최종: 줄만 읽어 썼다. 하위 유형별 형식 실패 수와 '열 개 가운데 아홉'은 같은 문서 § 2.3의 사후 기술(사전 등록 아님)이다.
- 10 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.3의 사후 기술(사전 등록 아님).
- 11 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 원출력 data/calls.jsonl seq 388(첫 실행)·seq 975(두 번째 실행), 질문 t1c-122. 미리 지정한 40개의 목록은 data/sets/designated.json.
- 12 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.1·§ 2.4 · 세 번째 시연(t1d-03) 다섯 번째 실행의 원출력 data/calls.jsonl seq 33. 입력 부트스트랩 10,000번. 반복

실행은 출력 분산에만 쓰고 실패율에 넣지 않는다(data/prereg.md § 5). temperature는 공급자 기본값이다 (같은 문서 § 9).

- 13 James A. Hanley-Abby Lippman-Hand, 1983-04-01, "If Nothing Goes Wrong, Is Everything All Right? Interpreting Zero Numerators", JAMA 249(13):1743-1745, doi:10.1001/jama.1983.03330370053031, Comment 절 — "it often seems to have a qualitative impact far in excess of its quantitative meaning." · "a new diagnostic test that has not yet misclassified a patient, a still-perfect surgical record ... an airline that has never had a fatality". 0의 무게에 대한 설명은 저자들의 해석이지 이 글의 실험 결과가 아니다. ↩
- 14 같은 글 「Making Inferences: The Rule of Three」 절 — "if none of n patients shows the event about which we are concerned, we can be 95% confident that the chance of this event is at most three in n". 두 사람이 단 글은 Rümkle, NEJM 1975-02-13, 292(7):372-373, "Implications of the Statement: No Side Effects Were Observed". ↩
- 15 같은 글 Table 2와 그 괄호 단락 — "the maximum risk [ie, 26% if n=10] is becoming so high that the difference between it and that suggested by the rule [3/10=30%] is not usually worth quibbling about." 30.8%(양쪽 정확 구간)·23.8%(균등 사전분포 베이지)·91.7%(라플라스 계승 규칙, 11/12)·34.9%·59.9%·90.4%(0.9·0.95·0.99의 10제곱)는 표준 이항·베타 계산이다. 'no problems so far'는 글의 끝 문단에 있다. ↩
- 16 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 시연 선택 규칙 data/prereg.md § 4, 실패 위치 data/m1-m2-results.md § 2.3·§ 4. 0/10의 25.9%는 독립·동일 분포 시행을 전제한 계산이다 (data/prereg.md § 5). ↩
- 17 FDA CDER, 2005-02-28(2005-03-03 갱신), "FDA Public Health Advisory: Suspended Marketing of Tysabri (natalizumab)", <http://web.archive.org/web/20050409081137/http://www.fda.gov:80/cder/drug/advisory/natalizumab.htm> — "one confirmed, fatal case and one additional case" · "At the time of approval, approximately 1,100 patients with MS had received Tysabri for one year or more." · "No cases of PML were observed during the clinical trials performed prior to approval of Tysabri." ↩
- 18 FDA, 2004-11, TYSABRI 라벨, Drugs@FDA, https://www.accessdata.fda.gov/drugsatfda_docs/label/2004/4125104lbl.pdf — "A total of 1,617 multiple sclerosis patients ... have been exposed to TYSABRI® with a median duration of exposure of 20 months." 0.185%와 0.272%는 $1 - 0.05^{1/n}$ 에 $n = 1,617$ 과 1,100을 넣은 값이다. ↩
- 19 Tarek A. Yousry 외, 2006-03-02, "Evaluation of Patients Treated with Natalizumab for Progressive Multifocal Leukoencephalopathy", NEJM 354(9):924-933, <https://pubmed.ncbi.nlm.nih.gov/16510746/> — "Only the three previously reported cases of PML were confirmed (1.0 per 1000 treated patients; 95 percent confidence interval, 0.2 to 2.8 per 1000)." · "The risk associated with longer treatment is not known." 분모는 다발성 경화증·크로병·류마티스 관절염 환자를 합한 것이라, 다발성 경화증 1,617명에서 쉼한 0.185%와의 비교는 어렵다. 19.8%는 0.999의 1,617제곱이다. ↩
- 20 FDA, 나탈리주맙 정보 페이지(2006-10-09 보존본), <http://web.archive.org/web/20061009231717/http://www.fda.gov/cder/drug/infopage/natalizumab/> — "withdrawn by the manufacturer in February 2005". 영구 철회가 아니라 제조사의 자진 판매 중단이었고, 2006-06-05 TOUCH 조건으로 재개가 승인됐다. ↩
- 21 ICH, 1994-10-27, "The Extent of Population Exposure to Assess Clinical Safety for Drugs Intended for Long-Term Treatment of Non-Life-Threatening Conditions (E1)", https://database.ich.org/sites/default/files/E1_Guideline.pdf — "When no serious ADE is observed in a one-year exposure period this number of patients can provide reasonable assurance that the true cumulative one year incidence is no greater than 3%." · "previous experience with the occurrence and detection of adverse drug events (ADEs), statistical ↩

considerations of the probability of detecting specified frequencies of ADEs, and practical considerations". 39%와 9.5%는 $1 - (1 - p)^{100}$ 에 $p = 0.5\%$ 와 0.1% 를 넣은 값이다.

- 22 같은 지침 — "Japan currently accepts 500–1500 patients: the potential for a smaller number of patients is due to the post-marketing surveillance requirement" · "It is anticipated that the total number of individuals ... will be about 1500." 77.7%와 13.9%는 $1 - (1 - p)^{1,500}$ 에 $p = 1/1,000$ 과 $1/10,000$ 을 넣은 값이다. ↩
- 23 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 과업 정의 data/prereg.md § 2, 결과 data/m1-m2-results.md § 1.2· § 2.2· § 2.3· § 2.4. 1차 모델의 실패 여섯은 변경 전 날짜 3, 취소된 예전 예약번호 1, 변형 표기 예약번호를 7자로 옮김 1, 휴대폰 번호의 하이픈 위치 1이다. ↩
- 24 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.2· § 2.3· § 2.4 · 예측과 근거 data/prereg.md § 6. 변형 표기 예약번호의 예는 세트 data/sets/t2_challenge.jsonl의 질문 t2c-135·t2c-199. ↩
- 25 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 기준 data/prereg.md § 7 · 두 예측 같은 문서 § 6 · 판정과 예측 대조 data/m1-m2-results.md § 1.1· § 1.2. ↩
- 26 Klarna Group plc, 2025-09-02, Form F-1/A Amendment No. 3, SEC EDGAR, <https://www.sec.gov/Archives/edgar/data/2003292/000200329225000024/klarnagroupplcf-1a3.htm> — "Our AI assistant handled 69% of customer service chats in the last twelve months ended June 30, 2025, according to our service chat log data, doing the work equivalent of over 700 full-time agents", 첫 달 수치는 Klarna 보도자료(2024-02-27, <https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/>). 회사 자체 측정이고, 700명은 해고 인원이 아니라 업무를 완산한 추정치다. ↩

채점자를 잔다

회계 책임자

1980년 뉴욕 대학 컴퓨터과학과의 기술보고서 25호에서 Elaine Weyuker는 프로그램 시험이라는 일의 밑바닥에 깔린 가정 하나를 꺼냈다. 시험을 연구하는 사람들은 프로그램의 출력이 맞는지 틀렸는지를 누군가 가려낼 수 있다고 전제한다. 그 누군가를 오라클이라 부른다. 사람이 오라클일 때도 있고, 정답표나 다른 프로그램이 그 자리에 설 때도 있다. Weyuker는 그 전제가 무너지는 경우를 이렇게 정의했다. "오라클이 없거나, 출력이 맞는지 판정하는 데 시험하는 사람이 비당한 시간을 들여야 한다면, 그 프로그램은 시험할 수 없다."¹

정답을 몰라도 틀린 답은 가려낼 수 있다는 것을 보이려고 그가 든 예는 큰 회사의 총자산을 계산하는 프로그램이다. 출력이 100달러나 1,000달러라면 그럴듯한 답이 아니다. 1,000,000달러는 시험하는 사람에게는 그럴듯해 보여도, 회사의 회계 책임자라면 틀렸다고 가려낼 만한 정보가 있을 수 있다. 그러나 회계 책임자도 1,134,906.43달러가 맞고 1,135,627.85달러가 틀렸다는 것까지 쉽게 가려내기는 어렵다. Weyuker는 이렇게 썼다. "그래서 전문가조차 그럴듯한 오답을 정답으로 받아들일 수 있다."²

회계 책임자는 그래도 쓸모가 있다. 말이 안 되는 숫자를 걸러 범위를 좁힌다. Weyuker는 같은 식으로 $\sin 42^\circ$ 의 값을 0.6657과 0.7071 사이로 좁히는 예를 들었다. 정답을 몰라도 범위 밖의 답은 틀렸다고 말할 수 있는 이런 판정자를 부분 오라클이라 부른다. 그가 꼽은 시험할 수 없는 프로그램의 첫 부류는 답을 알아내려고 짠 프로그램이다. 정답을 알았다면 그 프로그램을 짤 필요가 없었다. 다른 부류도 있었다. 30자리 숫자가 줄줄이 찍힌 출력 한 쪽은 사람이 대조하기에 그저 너무 지루할 수 있다. 시험하는 사람이 원래 문제와 다른 명세를 기준으로 출력을 보고 있을 수도 있다. 결론의 마지막 문장은 이렇다. "이것이 시험하는 사람이 마주해야 할 근본적인 한계라고 우리는 믿는다."³

2015년, 1978년부터 2012년까지의 출판물 694편을 정리한 소프트웨어 시험 서베이는 '시험 오라클'이라는 말이 1978년 William Howden의 글에서 처음 나왔다고 적고,

자동으로 만든 오라클이 모두 모자랄 때 마지막 판정자는 여전히 사람이라고 정리했다. 그 사람이 회계 책임자라면 판정은 범위를 좁히는 데서 멈춘다. 정답 없는 출력 앞에서 판정자를 믿을지 버릴지만 남는 것은 아니다. 판정자도 시험할 수 있다. 같은 질문을 다시 묻거나 순서를 바꿔 물어 판정이 흔들리는지 보고, 정답을 아는 문제를 섞어 넣어 맞는지 보면, 그 판정자가 무엇을 가리고 무엇을 못 가리는지가 숫자로 나온다.⁴

결항 승객

누리별항공 환불 앱에도 오라클이 없는 자리가 있다. 한 질문은 이렇다. "구매한 지 25일 됐어요. 스탠다드 표로 총 252,000원 결제했는데 그중 세금·공항이용료가 58,000원이에요. 항공사에서 그 편이 결항됐다고 연락이 와서 안 가려고요. 원래 출발 시각까지 6시간 남았어요. 환불 얼마 받을 수 있나요?" 규칙 엔진의 답은 252,000원이다. 항공사가 결항하면 결제 총액 전액이고, 그 조항이 다른 조항보다 먼저 온다.

claude-haiku-4-5의 답은 상황 분석, 적용 규정, 계산의 세 칸으로 나뉜다. 구매 경과 25일과 남은 6시간, 운임 194,000원을 차례로 적고, 제3조의 적용 순서에 따라 제4조가 먼저라고 밝힌 뒤 "구매 경과 시간이나 남은 시간과 관계없이" 전액이라고 맺는다. gpt-5.5의 답은 두 문장과 계산 한 줄이다. "항공사 사유로 결항되어 탑승하지 않고 취소하는 경우에는 제4조가 먼저 적용됩니다. 따라서 구매 후 24시간 초과 여부나 출발까지 6시간 남은 점, 스탠다드 수수료와 관계없이 결제 총액 전액 환불입니다." 두 답의 마지막 줄은 같다. "최종: 환불액 252000원."⁵

규칙 엔진은 두 답에 똑같이 정답 표시를 한다. 설명의 길이와 순서, 조항을 옮기는 방식은 규칙 엔진이 읽는 칸이 아니다. 240개의 설계가 정한 것은 금액과 「판단 불가」 뿐이고, 어느 안내문이 나온지는 정하지 않았다. 그래서 거기에는 규칙 엔진이 없다. 두 모델의 답이 모두 맞은 질문들 가운데 100개를 뽑아 이런 쌍을 만들었다. 결론은 같고 설명만 다른, 정답이 하나가 아닌 쌍이다. 이 측정은 이런 쌍을 열린 쌍이라 불렀다. 여기서 판정자 노릇을 할 수 있는 것은 사람이나 다른 모델이다.⁶

판정자들

1953년 색인 체계 둘을 맞대 보려던 두 팀은 어떤 문서가 질문에 '관련'되는지 합의하지 못해 공동 결과를 내지 못했다.⁷ 그 일을 1962년 보고서에 옮긴 크랜필드의 Cyril Cleverdon은 관련성 논쟁에 빠지지 않으려고 질문을 컬렉션 안의 문서에서 짓게 했다. 검색이 그 '출처 문서'를 찾으면 성공이었으니, 정답은 질문이 태어날 때 이미 정해져 있었다.⁸ Sparck Jones가 1981년 결산에 옮겨 적은 대로라면, 1965-01 Don Swanson은 출처 문서로 지은 질문과 관련성을 통제하지 않은 점 등을 들어 "설계 자체가 발견된 결과의 상당수를 보강한 것으로 보인다"고 썼다.⁹ 그 앱의 240개도 질문과 정답을 한 설계가 함께 만들었지만, 실패가 어느 층에 얼마나 쌓일지는 출력이 보여 졌다.

1992년에 시작된 검색 평가 대회 TREC에서는 참가한 시스템이 모두 같은 문서와 주제(질문)로 같은 채점을 받았다. 판정은 문서 전부가 아니라 시스템마다 상위에 올린 문서를 모은 풀에만 내려졌고(폴링), 풀에 들지 못한 문서는 판정 없이 비관련으로 섰다.¹⁰

규칙 엔진은 같은 사실에 늘 같은 판정을 낸다. 판정자가 사람이면 같은 기준을 받아도 판정이 같린다. TREC-4에서는 49개 주제를 주제마다 세 사람이 따로 판정했고, Voorhees가 그 결과를 맞대 봤다. 관련 문서 집합이 겹친 정도, 곧 둘 다 관련이라고 본 문서를 둘 중 하나라도 관련이라고 본 문서로 나눈 값의 주제 평균은 두 사람 사이에서 0.421~0.494, 세 사람 모두에서 0.301이었다.¹¹

그런데 판정을 바꿔도 시스템 순위는 거의 그대로였다. 원래 판정으로 매긴 순위와의 Kendall 상관은 셋 중 하나라도 관련이라고 본 문서를 관련으로 치면 0.9508, 셋 모두 관련이라고 본 문서만 치면 0.9015였다. Kendall 상관은 두 순위가 같은 순서일수록 1에 가까워지는 값이다. 주제마다 세 판정자 가운데 하나를 무작위로 골라 짤 판정 집합들로 매긴 순위도 원래 순위와의 상관이 평균 0.938이었다. 집합끼리 판정을 일부 나눠 가져 이 값은 조금 높게 나왔을 것이라고 Voorhees는 적었다. 그의 결론은 이렇다. "완벽한 검색을 정의하는 데 쓰인 관련성 판정이 크게 달라도, 검색 전략들의 상대적 효과는 안정적이다." 단서도 붙였다. 관련 문서가 다섯 건 안팎보다 적은 질의에서는 점수 자체가 불안정하고, 사용자와 많이 주고받는 시스템은 더 흔들리며, 서로 다른 시스템끼리 견줄 때는 한 시스템의 변형끼리보다 더 큰 점수 차가 있어야 의미가 있다. 절대 점수는 판정에 따라 움직였고, 순위가 버틴 시스템들은 1995년 TREC-4 애드혹 과제에 전체 문서로 결과를 낸 33개였다.¹²

Cleverdon도 1970-10의 보고서에서 Cranfield II의 색인 언어 순위가 판정에 휘둘렸는지 확인했다. 보고서는 Cranfield II의 주된 시험들이 원래 문서와 질문의 부분 모음, 곧 문서 200건과 질문 42개 위에서 이뤄졌다고 적고, 그 부분 모음을 다시 썼다. 다른 세 사람의 새 판정으로 19개 색인 언어를 다시 셈하자, 세 순위 모두 원래 순위와의 상관이 0.921 이상이었다.¹³ 판정이 갈려도 남은 안정은 여러 질문에 걸쳐 평균 낸 시스템 순위의 것이었고, 안내문 하나하나의 우열은 재지 않았다.

심판

모델을 판정자로 세운 연구 가운데 한 갈래는 답 하나하나에 내린 판정을 잼다. 2023-06 Lianmin Zheng 외 12명은 챗봇의 답 두 개를 놓고 GPT-4 같은 모델에게 어느 쪽이 나은지 묻는 방식을 시험했다. 두 답의 순서만 바뀌도 같은 판정이 나오는 비율은 Claude-v1이 23.8%, GPT-3.5가 46.2%, GPT-4가 65.0%였다. "60%가 넘는 경우에 일관된 결과를 낸 것은 GPT-4뿐"이었다. Claude-v1은 기본 프롬프트에서 75.0%를 첫 번째 답 편으로 판정했고, 답의 이름만 바꾼 프롬프트에서는 "Assistant A"라는 이름을 편들었다. 논문이 실은 한 예에서 GPT-4는 GPT-3.5의 답이 앞에 오면 GPT-3.5를, 뒤로 보내면 Vicuna를 골랐다. 저자들은 비교한 답들이 매우 비슷해 사람에게도 어려운 시험이었다는 단서를 달았다. 2023년의 모델들이 낸 숫자다.¹⁴

수학 문제에서는 다른 약점이 보였다. 문제 10개를 순서를 바꿔 20번 판정하게 하자, GPT-4가 틀린 답을 맞다고 한 경우가 기본 프롬프트로 14번, 단계별로 생각을 쓰게 하면 6번, 판정 전에 GPT-4가 따로 푼 답을 참고 답안으로 넣어 주면 3번이었다. 따로 물으면 풀 수 있는 문제에서도 GPT-4는 제시된 답에 이끌렸다고 저자들은 적었다. 참고 답안은 판정자에게 바깥에서 들어온 오라클이다. 그것이 있을 때 판정이 나아졌다.¹⁵

같은 논문은 반대쪽 숫자도 냈다. 질문 80개에 모델 6개의 답을 놓고, 대부분 대학원생인 전문가 판정자 58명이 약 3,000표를 던졌다. 대화의 첫 차례에서 이들의 판정과 GPT-4의 판정을 맞대자, 무승부를 뺀 판정에서 GPT-4와 사람의 일치는 85%로 사람끼리의 81% 보다 높았다. 무승부를 넣으면 66% 대 63%였고(이어 묻은 둘째 차례에서는 66% 대 67%), 우연히 맞을 기준선은 무승부를 빼면 50%, 넣으면 33%다. 사람끼리 81%라는 숫자는 기준으로 세운 사람 판정자도 하나의 판정자라는 뜻이기도 하다. 자기 선택이 GPT-4와

갈린 판정에서 GPT-4의 판정을 보여 주자, 사람들은 그 가운데 75%에서 그 판정이 타당하다고 봤고 34%에서는 자기 선택을 바꿀 뜻이 있었다.¹⁶

모델의 순위로 올라가면 사람 순위와의 상관은 더 높게 나온다. 2024년 Dubois 외는 GPT-4 Turbo 자동 채점이 긴 답을 좋아하는 경향을 회귀로 보정하자, Chatbot Arena의 사람 순위와의 Spearman 상관이 0.94에서 0.98로 올랐다고 보고했다. 보정 전의 자동 채점은 답을 길게 쓰라는 지시만으로도 점수가 크게 흔들렸고, 저자들은 그것이 AI 시스템에게 상당히 공략당할 수 있다고 적었다. 같은 해 프롬프트 500개로 모델 하나를 20달러에 채점한다는 Arena-Hard는 초록에서 사람 선호 순위와 98.6% 상관을 내세웠다. 상위 20개 모델 전체를 비교한 표에서는 93.2%였고, 98.6%는 문제를 보정하고 어려운 프롬프트로 비교한 조건의 값이다. 같은 표에서 MT-Bench는 상관이 89.9%인데, '확신 일치(confidence agreement)'는 26.6%였다. 두 순위가 모두 확신을 갖고 가른 모델 쌍에서 순서가 같으면 1점, 다르면 -1점, 어느 한쪽이라도 가르지 못하면 0점을 매긴 값이다. 판정자는 쓸모없지 않다. 모델의 순위를 맞히는 일과 답 하나하나를 맞히는 일이 다를 뿐이다.¹⁷

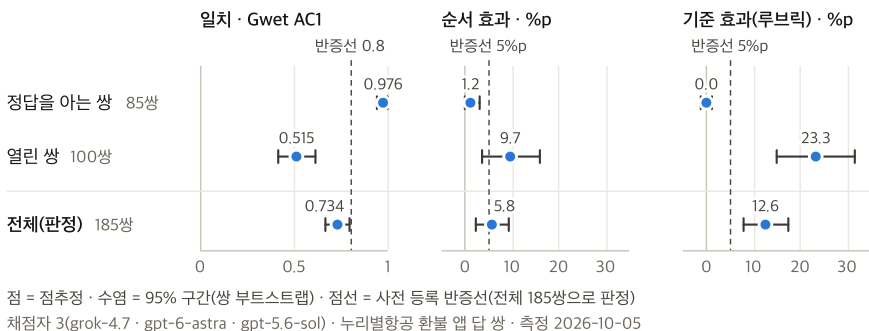
세 채점자

누리별항공 환불 앱의 답 쌍으로 채점자를 잤다. 측정의 꼴은 TREC-4의 세 판정자 실험을 채점자에게 옮겨 놓은 것과 같다. 거기서 이어받는 것은 판정자를 바꾸면 비교가 남는가라는 분석 질문 하나다. TREC-4가 그 질문으로 잰 것은 순위의 안정이었고, 이 측정은 순위를 만들지 않았다. 쌍을 짓는 법과 채점 기준, 순서 교환과 재질문은 이 측정의 절차로 따로 정했다. 사전 등록은 쌍을 두 묶음으로 나눴다. 하나는 정답을 아는 쌍 85개다. 두 모델 가운데 한쪽만 맞힌 질문 27개(그런 질문 37개 가운데 틀린 답이 형식 실패인 10개는 뺐다), 그리고 숫자만 바꾼 가짜 규정을 claude-haiku-4-5에게 주어 받아 낸 확신에 찬 오답을 원래의 정답과 짝지는 58개다. 다른 하나는 결항 승객의 두 답 같은 열린 쌍 100개다. 판정은 grok-4.7, gpt-6-astra, gpt-5.6-sol 세 채점자가 맡았다. 규정 원문과 채점 기준을 주고, 어느 모델이 쓴 답인지는 가렸다. 기준은 결론이 규정에 맞는지를 먼저, 조항과 계산의 설명이 정확한지를 다음으로, 고객이 이해하기 쉽고 정중한지를 마지막으로 보라고 했다. 우열을 가릴 수 없으면 동점이고, 답의 길이와 제시 순서는 기준이 아니라고 적었다. 쌍마다 네 번 물었다. 원래 순서로, 순서를 바꿔, 원래 순서로 한 번 더, 그리고 첫

결과를 본 뒤 모호한 곳을 고친 기준으로, 판정은 모두 2,220개였고, 판정 줄을 읽지 못한 출력은 없었다.¹⁸

반증선은 측정 전에 정했다. 순서를 바꿔 움직인 판정, 곧 순서 효과가 5%p 미만이고, 세 채점자의 일치가 0.8을 넘고, 기준 문구를 고쳐 움직인 판정, 곧 기준 효과가 5%p 미만이면, 이 설정에서는 채점자가 치우친다고 주장할 근거가 모자란다고 보고하기로 했다. 움직인 판정은 같은 순서로 다시 물었을 때 바뀐 비율을 빼고 썼다. 다시 묻기만 해도 바뀌는 만큼은 순서나 기준의 탓이 아니기 때문이다. 일치는 처음에 Fleiss κ 로 재기로 했다가, 결과보다 먼저 Gwet의 AC1로 바꿨다. κ 는 판정이 맞아떨어진 정도에서 우연히 맞아떨어진 만큼을 덜어 낸 값이고, 1이면 완전한 일치다. AC1도 우연히 같은 만큼을 덜어 내지만, 판정이 한 범주로 몰릴 때 관측 일치가 높은데도 값이 낮게 나오는 κ 의 역설을 덜었다.¹⁹

정답을 아는 쌍에서 세 채점자는 98.8%를 맞혔다. 동점도 오답으로 센 값이다. 가짜 규정으로 받아 낸 오답 58쌍은 세 채점자 모두 하나도 놓치지 않았고, 두 모델 가운데 한쪽만 맞힌 27쌍에서는 96.3%였다. 일치는 AC1 0.976, 순서를 바꿔 움직인 판정은 1.2%p였다. 기준 문구를 고치자 바뀐 판정은 3개(1.2%)였는데, 같은 순서로 다시 물었을 때 바뀐 비율과 같아 기준 효과는 0.0%p다. 같은 85쌍으로 세 채점자의 Fleiss κ 를 셈하면 -0.012가 나온다. 세 판정의 관측 일치가 0.976인데 그렇다. 판정이 정답 쪽으로 몰릴수록 κ 가 0 근처로 떨어지는 역설이 실측에서 그대로 나타났다.²⁰



세 채점자(grok-4.7·gpt-6-astra·gpt-5.6-sol)가 2026-10-05에 판정한 누리별항공 환불 앱 답 쌍 185개(정답을 아는 쌍 85·열린 쌍 100)의 일치(Gwet AC1)·순서 효과·기준 효과와 쌍 부트스트랩 95% 구간이며, 점선은 측정 전에 등록한 반증선이다.

열린 쌍에서는 달랐다. AC1은 0.515로 떨어졌다. 순서를 바꾸면 판정의 27.0%가 바뀌었고, 같은 순서로 다시 물어도 17.3%가 바뀌었다. 그 차이인 순서 효과가 9.7%p다. 기준 문구를 고치자 판정의 40.7%가 움직였고, 재검사 변동을 빼면 23.3%p다. 판정의 44.7%가 동점이었다. 185쌍 전체로는 순서 효과 5.8%p(95% 구간 2.3~9.4%p), AC1 0.734(0.669~0.798), 기준 효과 12.6%p(7.9~17.5%p)였다. 세 항 모두 반증 조건을 채우지 못했고, 셋 다 채점자가 치우친다는 쪽에 놓였다. 순서 효과는 점추정이 반증선을 0.8%p 넘었고 구간이 반증선을 걸쳐, 셋 가운데 가장 얇은 근거다. 나는 일치와 정확도를 둘 다 더 낮게 예측했다(185쌍의 AC1 0.53, 85쌍의 AC1 0.78과 정확도 88%). 채점자들은 예측보다 정확했고 서로 더 일치했다.²¹

결항 승객의 쌍에서 세 채점자의 판정은 셋으로 갈렸다. grok-4.7은 긴 답을 골랐다. 제3조의 적용 순서와 운임·세금 구성을 함께 밝혀 계산 근거가 더 분명하고, 응대도 더 정중하고 이해하기 쉽다는 이유였다. gpt-5.6-sol은 짧은 답을 골랐다. "결항되어 탑승하지 않고 취소하는 경우"라는 제4조의 적용 조건까지 더 정확히 설명했다는 이유였다. gpt-6-astra는 우열을 가리기 어렵다며 동점을 줬다. 두 답의 순서를 바꾸자 gpt-5.6-sol의 판정은 동점으로 옮겨 갔다.²²

채점자마다 결도 달랐다. 순서 효과는 grok-4.7이 3.2%p, gpt-6-astra가 6.5%p, gpt-5.6-sol이 7.6%p였고, 동점을 준 비율은 16.2%, 32.4%, 25.4%였다. 기준 효과는 거꾸로 grok-4.7이 15.7%p로 가장 컸다. 합쳐 보면 첫 번째 자리의 답을 고르는 실패는 없었다. 동점을 뺀 판정에서 세 채점자가 A를 고른 비율은 45~49%였다. 순서 효과는 늘 앞의 답을 고르는 꼴이 아니라 순서를 바꾸면 판정이 바뀌는 꼴로 나타났다.²³

채점자 구성에도 단서가 붙는다. 사전 등록한 세 번째 채점자는 claude-sonnet-4-6이었다. 측정 당일 이 모델은 상류의 레이트 리밋 때문에 한 시간 넘게 응답을 내지 못했고, 사전 등록한 단 한 번의 교체 경로대로 gpt-5.6-sol로 바꿨다. 그래서 OpenAI 계열이 둘이 됐다. 같은 공급자의 모델은 판정 습관이 비슷할 수 있으니 일치가 위로 치우쳤을 수 있다. 측정된 AC1은 이미 0.8 아래라, 치우침이 있었다면 실제 일치는 더 낮다.²⁴

열린 쌍에서 세 채점자가 보인 AC1 0.515를 판정자들의 겹침이나 순위 상관 옆에 놓을 수는 있지만 같은 양으로 읽을 수는 없다. 겹침과 순위 상관은 관련 문서 집합과 시스템의 순서를 재고, AC1은 쌍 하나하나에 내린 판정이 우연 이상으로 얼마나 같은지를 잴다. 이

측정은 두 모델의 순위를 재지 않았다. 판정은 쌍마다 받았을 뿐 그 판정들을 모아 순위를 세우지 않았으니, 판정이 갈려도 순위가 버틴다는 결과를 전줄 자리가 이 측정에는 없다.

이 측정이 보여 주는 것은 더 좋다. 같은 열린 쌍 100개를 원래 순서로 처음 물은 판정에서 gpt-5.5의 답을 고른 수, claude-haiku-4-5의 답을 고른 수, 동점의 수는 grok-4.7이 32·38·30, gpt-6-astra가 33·10·57, gpt-5.6-sol이 36·17·47이었다. 어느 모델의 안내문이 나온가에 대한 답은 채점자에 따라 달랐다. 둘은 gpt-5.5의 답을 훨씬 많이 골랐다. 하나는 claude-haiku-4-5의 답을 조금 더 골랐지만(38 대 32) 어느 쪽으로도 뚜렷하지 않았다. 동점을 준 수가 채점자마다 달라서 비율을 내면 분모가 다르고, 이 수는 첫 판정만 센 것이다. 채점자마다 claude-haiku-4-5의 답을 고른 비율은 사전 등록에서 선호로만 보고하기로 정한 값이지만, 세 수를 이렇게 펴친 표시와 Wilson 구간은 결과를 본 뒤 더한 사후 기술이다. gpt-6-astra와 gpt-5.6-sol은 gpt-5.5와 같은 OpenAI 계열이다. 어느 모델의 답인지는 가렸으니 이 수는 선호로만 적는다. 같은 계열의 답을 더 고른 선호인지 안내문의 품질 차이인지는 이 측정으로 가르지 못한다.²⁵

문지기

2023-05-25 뉴욕 남부 연방지방법원에 진술서 하나가 들어왔다. Mata v. Avianca 사건에서 원고 측 변호사 Steven Schwartz가 서명한 것이다. 30년 넘게 주 법원 사건만 맡아 온 그가 ChatGPT로 법률 조사를 한 것은 이 사건이 처음이었다. ChatGPT는 대학생 나이의 자녀들에게서, 그리고 읽은 기사들에서 알게 됐다고 그는 뒤에 적었다. 진술서에 붙은 화면은 휴대폰 캡처다. 상태 표시줄의 시각은 3:43, 배터리는 충전 중이고, 대화 제목은 "Tolling Montreal Convention Statute"다. 그는 ChatGPT에게 "Is varghese a real case"라고 물었다. 답은 실제 판례라는 것이었다. "What is your source"라고 되물자 답은 사과로 시작했다. "앞서의 혼란에 사과드립니다. 다시 확인해 보니 Varghese 사건은 실제로 존재합니다." 그가 다시 "Are the other cases you provided fake"라고 묻자 답은 이랬다. "아니요, 제가 드린 다른 판례들도 진짜이며 LexisNexis나 Westlaw 같은 믿을 만한 법률 데이터베이스에서 찾을 수 있습니다."²⁶

판례가 진짜인지를 그 판례를 내놓은 기계에게 다시 물은 것이다. 이 질문을 언제 했는지에 대한 그의 설명은 바뀌었다. 5월의 진술서는 서면을 내기 전에 물은 것처럼 읽혔고, 6월의

선언서는 법원이 2023-05-04 명령을 낸 뒤 이미 크게 의심하던 상태에서 물었다고 했다. 법원은 이를 이랬다저랬다 하는 모순된 설명이라 불렀다.²⁷

바깥의 확인은 일찍 왔다. 서면을 낸 지 2주 만인 2023-03-15 상대방 답변서가 인용된 판례 대부분을 찾을 수 없다고 적었고, 2023-04-11과 2023-04-12에는 법원이 판례 사본을 내라고 명령했다. 두 변호사는 물려서지 않았다. 2023-06-08 정오, 펠 스트리트 500번지 법원의 11D호 법정에서 열린 심리에서 판사가 물었다. "Varghese 판결 전문을 찾아보러 갔을 때 무엇을 찾았습니까?" "찾지 못했습니다." "그런데도 그것을 내게 낸 서면에 인용했군요." 그는 Google에서는 찾을 수 없는 판례도 있으리라 여겼다고 답했다. 사무소가 쓰던 Fastcase로는 연방 판례를 볼 수 없었고, Westlaw나 Lexis 계정은 없었다. ChatGPT는 슈퍼 검색 엔진 같은 것이라고 잘못 짐작했다고 그는 말했다. 판사가 가짜 Varghese 판결의 한 대목을 두고 법률적으로 황실수설이라는 데 동의하느냐고 묻자, 그는 지금 보니 그렇다고 답했다. 서면에 서명한 동료 변호사 LoDuca는 법정에서 자기는 기본적으로 글의 흐름을 봤다고 말했다. Schwartz의 변호인 Minkoff는 Schwartz가 한 일을 실탐을 갖고 논 것이라고 불렀다.²⁸

반대서면이 들어간 것은 2023-03-01이다. 2023-05-04 법원은 명령의 첫 문장에 "법원은 전례 없는 상황을 마주했다"고 적었고, 2023-06-22 P. Kevin Castel 판사는 제재 의견서를 냈다. 두 변호사와 사무소에 연대로 5,000달러, 그리고 원고 Roberto Mata와 가짜 판결에 이름이 쓰인 판사들에게 편지를 보내라는 명령이었다. 의견서는 도구를 탓하지 않았다. "기술의 진보는 흔하고, 믿을 만한 인공지능 도구를 도우에 쓰는 일에 본래 부적절한 것은 없다. 그러나 현행 규칙은 변호사에게 제출물의 정확성을 지키는 문지기 역할을 지운다." 세 채점자의 판정은 순서를 바꾼 재질문과 정답을 아는 쌍으로 시험받았다. Schwartz의 확인에는 그런 장치가 없었다. 바깥의 신호는 2주 만에 왔지만, 판정은 서면을 낸 지 넉 달 가까이 지나 제재 의견서로 왔다.²⁹

정답을 아는 85쌍에도 놓친 판정이 있었다. gpt-6-astra는 개인 사정 질문 세 쌍에서 「판단 불가」로 끝낸 답과 금액에 단서를 단 답을 동점으로 봤다. 가족상이나 입원을 이유로 수수료 면제를 묻는, 규칙 엔진이 판단 불가라고 답하는 질문들이다. 같은 기준을 받은 grok-4.7과 gpt-5.6-sol은 원래 순서의 첫 판정에서 세 쌍 모두 판단 불가 쪽을 골랐다. 결론은 마지막 「최종:」 줄이고, 참고 금액을 최종 환불액으로 단정하면 결론이 틀린 것이라고 더 분명히 적은 기준에서 gpt-6-astra의 그 세 판정은 정답 쪽으로 옮겨

갔다. 정확도는 원래 순서의 첫 판정으로 섰고, 동점은 오답으로 섰다. 그렇다고 그 판정에서 gpt-6-astra가 틀린 결론을 통과시킨 것은 아니다. 두 답 가운데 고르지 않았을 뿐이다. 그러나 세 쌍 가운데 가족상 질문 한 쌍을 순서를 바꿔 물었을 때와 같은 순서로 다시 물었을 때, gpt-6-astra는 금액에 단서를 단 쪽을 골랐다. 순서를 바꾼 그 판정에서는 gpt-5.6-sol도 같은 쪽을 골랐다. 다른 가족상 질문 한 쌍에서는 순서를 바꾸자 세 채점자가 모두 금액에 단서를 단 답을 골랐다(결과를 본 뒤 쌍을 열어 본 사후 기술이다).³⁰

채점자를 켜 숫자는 한 줄로 서지 않는다. 일치하는 세 채점자가 서로 같은 판정을 냈는지를 켜다. 정답을 아는 쌍에서 정답 쪽을 골랐는지는 정답 정확도가 켜다. 고르지 않고 동점을 준 자리는 판정 불가로 따로 남는다. 일치가 높아도 정답에서 멀 수 있고, 판정 불가는 오답과 같은 말이 아니다.

주

- 1 Elaine J. Weyuker, 1980, "On Testing Nontestable Programs", NYU Computer Science Department Technical Report No. 025, <http://web.archive.org/web/20221205071201/https://bbst.courses/wp-content/uploads/2022/08/Weyuker-ontestingnontestable.pdf>, 초록 — "A program is nontestable if either an oracle does not exist or the tester must expend some extraordinary amount of time to determine whether or not the output is correct." 표지의 날짜는 "OCTOBER 198"에서 끝자리가 잘렸고, 참고문헌의 가장 늦은 해는 1980이다. 1982년 학술지판(The Computer Journal 25(4):465–470)은 제목과 용어를 'Non-Testable'로 썼다. 인용은 1980년 보고서의 문구다. ↩
- 2 같은 보고서 § 2 — "It is unlikely that the comptroller can readily determine that \$1,134,906.43 is correct, and \$1,135,627.85 is incorrect." · "Thus even an expert may accept incorrect but plausible answers as correct results." ↩
- 3 같은 보고서 § 2와 결론 — "Programs which were written to determine the answer. If the correct answer were known, there would have been no need to write the program." · "A single output page containing columns of 30 digit numbers may simply be too tedious" · "This, we believe, is the fundamental limitation that testers must face." sin 42° 의 범위도 같은 절에 있다. 1980년의 수치 계산 예를 오늘의 문장 출력에 옮기는 것은 유비다. ↩
- 4 Earl T. Barr·Mark Harman·Phil McMinn·Muzammil Shahbaz·Shin Yoo, 2015–05(온라인 2014–11–19), "The Oracle Problem in Software Testing: A Survey", IEEE Transactions on Software Engineering 41(5):507–525, <https://discovery.ucl.ac.uk/1471263/1/06963470.pdf>, p.507·p.510 — "When none of these is completely adequate, the final source of test oracle information remains the human" · "The term "test oracle" first appeared in William Howden's seminal work in 1978", Howden의 1978년 글은 서베이를 거쳐 옮겼다. ↩
- 5 저자 측정(사전 등록 2026–10–05 10:32, 한국 시각 — 결과보다 먼저), 질문 t1c–148(data/sets/t1_challenge.jsonl), 두 답의 원출력 data/calls.jsonl seq 425(claude–haiku–4–5)·seq 800(gpt–5.5). ↩
- 6 저자 측정(사전 등록 2026–10–05 10:32, 한국 시각 — 결과보다 먼저), 쌍 구성 규칙 data/prereq.md § 8 · data/m2_pairs.py, 표본 data/m1–m2–results.md § 3.1. 두 답 가운데 하나라도 형식 실패인 질문은 쓰지 않았다. ↩
- 7 Cyril Cleverdon, 1991, "The significance of the Cranfield tests on index languages", SIGIR '91 Proceedings pp.3–12, <https://doi.org/10.1145/122860.122861>, p.5 — "at the end of the second day of meetings, they were still arguing about the meaning of the first search question." 1962년 보고서는 Cyril Cleverdon, 1962–10, Report on the Testing and Analysis of an Investigation into the Comparative Efficiency of Indexing Systems, Aslib Cranfield Research Project, <http://hdl.handle.net/1826/836> 이다. 그 2장 pp.7–8은 이 시험을 "an investigation known as the ASTIA–Uniterm test, which was made in 1953"이라 적고, "never fully written up"이라 일반에 볼 수 있는 기록은 Gull의 논문뿐이라고 밝혔다. 이 일은 Cleverdon의 기술에 기댄다. ↩
- 8 1962 보고서 2장 p.8 — "it should not get bogged down in the quagmire of arguments concerning relevancy" · pp.8–10 — "by using questions which were based on documents that were in the collection". 성공 기준은 p.13 — "If a card bearing the project code number of the source document was there, then the search was considered successful". ↩

- 9 Don R. Swanson, 1965-01, "The evidence underlying the Cranfield results", Library Quarterly 35(1):1 – 20, <https://www.jstor.org/stable/4305528> — "the design itself seems to have guaranteed many of the results that were found, so that the evidence which supports such results is questionable." · "The value of the project as a whole has been unquestionably great". 원문은 직접 대조하지 못했고, Sparck Jones, 1981, "The Cranfield tests", Information Retrieval Experiment pp.256 – 284, https://www-nlpir.nist.gov/project/sirlib/pubs/ire/ire_text/ 의 pp.268 – 270이 옮긴 문장을 따랐다. Cleverdon의 반박은 1991 p.5 — "His major point related to the use of search questions which were based on documents in the test collection." · "The validity of this point was somewhat lessened in that neither Swanson nor anyone else had been able to propose any other practical technique which would have overcome so effectively the problem that is associated with the determination of relevance". ↩
- 10 Voorhees, 2007(NIST 기록 2007-11-26), "TREC: Continuing Information Retrieval's Tradition of Experimentation", CACM 원고, https://tsapps.nist.gov/publication/get_pdf.cfm?pub_id=51229, p.1 — "started in 1992" · "information need statements called topics". TREC의 판정 절차는 p.2 — "TREC uses a process known as pooling [Sv75] in which the judge reviews only the documents in a topic's pool" · "assuming all unjudged documents are not relevant". 참가자가 같은 문서와 주제로 순위를 내고 NIST의 판정으로 채점받는 절차도 p.2다. ↩
- 11 Voorhees, 2000, "Variations in relevance judgments and the measurement of retrieval effectiveness", Information Processing & Management 36(5):697 – 716, http://www.jasonmorrison.net/iakm/cited/Voorhees_E_variations_in_relevance_judgements.pdf, § 3· § 3.1 pp.700 – 701·표 1 — "each topic was judged by three individuals" · "Overlap is defined as the size of the intersection of the relevant document sets divided by the size of the union of the relevant document sets." · "Table 1 gives the mean overlap". TREC-4의 주제 49개도 p.700이다. ↩
- 12 같은 글 pp.702 – 704·표 3· § 5 pp.714 – 715 — "the relative effectiveness of different retrieval strategies is stable despite marked differences in the relevance judgments used to define perfect retrieval." · "The study did detect circumstances in which system comparisons need to be done with more caution." 순위를 다시 매긴 시스템은 TREC-4 애드혹 과제의 category A 33개다(p.702). TREC-4의 해와 category A의 뜻은 Donna Harman, 1996-10, "Overview of the Fourth Text Retrieval Conference (TREC-4)", NIST Special Publication 500-236, <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication500-236.pdf> 를 따랐다. 회의는 1995-11에 열렸고(p.1), category A는 전체 문서로, B는 그 4분의 1로 참가한 쪽이다(p.3). ↩
- 13 Cyril Cleverdon, 1970-10, The Effect of Variations in Relevance Assessments in Comparative Experimental Tests of Index Languages, Cranfield Library Report No. 3, https://sigir.org/files/museum/The_Effect_of_Variations_in_Relevance_Assessments_in_Comparative_Experimental_Tests_of_Index_Languages/pdfs/front_matter.pdf, Summary — "It was desired to check whether the unexpected test results obtained in Cranfield II had been influenced by the relevance decisions." · "Three new sets of relevance decisions were therefore obtained, and the Cranfield II results were re-calculated for nineteen index languages" · "in no case did the correlation co-efficient of any of the three new rank orders fall below 0.921 when compared with the original Cranfield II results." 부분 모음은 p.3(같은 폴더의 p1.pdf, 쪽 이미지 기준) — "In Cranfield II, the main series of tests had been carried out on subsets of the original sets of documents and questions; these subsets, consisting of 200 documents and 42 questions ... were used in the present test." 원래 Cranfield II 순위도 같은 부분 모음의 것이다. 판정자 세 사람은 p.5이고, 상관은 Spearman 순위 상관이다(p.11). ↩
- 14 Lianmin Zheng 외 12명, 2023-06-09(v1: v4 2023-12-24), "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena", NeurIPS 2023 Datasets and Benchmarks Track, arXiv:2306.05685, <https://arxiv.org/abs/2306.05685>, § 3.3·표 2·그림 11 — "Only GPT-4 outputs consistent results in more than 60% of cases." ↩

- 15 같은 논문 § 3.3-3.4:표 4 — "A failure means when GPT-4 says an incorrect answer is correct." · "although GPT-4 can solve the problem (when asked separately), it was misled by the provided answers". 문제 10개, 판정 20회의 작은 표본이다. ↩
- 16 같은 논문 § 4.2:표 5(a 첫 차례·b 둘째 차례), 부록 C.1 — "The agreement under setup S2 (w/o tie) between GPT-4 and humans reaches 85%, which is even higher than the agreement among humans (81%)." · "when a human's choice deviated from GPT-4" · "reasonable in 75% of cases" · "willing to change their choices in 34% of cases". 사람에게 GPT-4의 판정을 보여 준 절차는 기준인 사람을 채점자 쪽으로 끌어당길 수 있다. ↩
- 17 Yann Dubois 외 3명, 2024-04-06(v2 2025-03-10), "Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators", COLM 2024, arXiv:2404.04475, 초록·§ 2·§ 4.2 — "it increases the Spearman correlation with LMSYS Chatbot Arena from 0.94 to 0.98" · "could be significantly gamed by AI systems" · Tianle Li 외, 2024-06-17(v2 2024-10-14), "From Crowdsourced Data to High-Quality Benchmarks: Arena-Hard and BenchBuilder Pipeline", arXiv:2406.11939, v2 초록·표 1·표 3 — "achieves 98.6% correlation with human preference rankings, all at a cost of \$20." v1 초록은 "89.1% agreement"와 25달러를 적었다. 둘 다 모델 순위 수준의 상관이고, 기준으로 쓴 Chatbot Arena의 순위도 하나의 판정이다. ↩
- 18 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 설계 data/prereg.md § 8, 채점 기준 원문 data/prompts.py(첫 기준)·data/m2_rubric_v2.txt(고친 기준), 표본과 조건 data/m1-m2-results.md § 3.1. ↩
- 19 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 반증선과 지표 정의 data/prereg.md § 8. κ 에서 AC1로 바꾼 것은 결과 전에 한 사전 등록 수정이고, κ 의 역설과 AC1에 대한 근거 문헌은 같은 절에 있다. ↩
- 20 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.2·§ 3.3(정답을 아는 85%). ↩
- 21 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 1.3·§ 3.2·§ 3.3(열린 쌍 100, 전체 185쌍). 구간은 쌍 단위 부트스트랩 10,000번이고, 예측은 data/prereg.md § 6. ↩
- 22 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 쌍 B-149(질문 t1c-148), 판정 원출력 data/calls.jsonl seq 2550(grok-4.7)·2185(gpt-6-astra)·3108(gpt-5.6-sol, 원래 순서)·3109(gpt-5.6-sol, 바꾼 순서). 이 쌍의 원래 순서에서 A는 gpt-5.5의 답이었다. ↩
- 23 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.5. ↩
- 24 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 교체 경로 data/prereg.md § 11(b), 실행 기록과 한계 data/m1-m2-results.md § 6·§ 7, claude-sonnet-4-6의 시도 10번은 2026-10-05 10:34~11:35에 모두 502로 끝났고, 이 모델에는 채점 요청을 한 건도 보내지 않았다. ↩
- 25 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저)의 원출력 data/calls.jsonl을 결과를 본 뒤 다시 쓴 사후 기술이다. 채점자별로 1차 모델(claude-haiku-4-5)의 답을 고른 비율은 사전 등록 data/prereg.md § 8이 「선호」로만 보고하기로 정한 기술 통계다. 고른 수와 동점 수를 채점자마다 이렇게 펼쳐 보이는 방식과 Wilson 구간은 사전 등록에 없다. 셸 data/m2_family_posthoc.py, 결과 data/m2-family-posthoc.md. 쌍 구성과 판정의 대응은 data/analyze.py의 것을 그대로 쓰고, data/results.json의 채점자별 값과 맞춰 봤다. 동점을 뺀 판정에서 gpt-5.5 답을 고른 비율과 Wilson 95% 구간은 grok-4.7 45.7% 34.6, 57.3, gpt-6-astra 76.7% 62.3, 86.8, gpt-5.6-sol 67.9% 54.5, 78.9이다. 구간은 채점자마다 따로 낸 기술 통계이고, 채점자 사이의 차이를 검정한 값이 아니다. 같은 비율은 data/m1-m2-results.md § 3.5에도 있다. ↩

- 26 Steven A. Schwartz, 2023-05-24 서명·2023-05-25 제출, Affidavit(ECF 32-1), Mata v. Avianca, Inc., 1:22-cv-01461(S.D.N.Y.), <https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.32.1.2.pdf>, 부속 화면(PDF 4-6쪽)·910 — 4쪽 "Yes, Varghese ... is a real case." · 5쪽 "I apologize for the confusion earlier. Upon double-checking, I found that the case Varghese ... does indeed exist" · "No, the other cases I provided are real and can be found in reputable legal databases such as LexisNexis and Westlaw." 화면은 발췌이고 날짜가 없다. 질문에는 원문대로 물음표를 붙이지 않았다. 경력과 ChatGPT를 알게 된 경위는 그의 6월 선언서 ECF 46 915-6·914. ↩
- 27 P. Kevin Castel, 2023-06-22, Opinion and Order on Sanctions(ECF 54), <https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.54.0.8.pdf>, Findings 9146-47 — "shifting and contradictory explanations". ↩
- 28 법원 속기, 심리 2023-06-08·제출 2023-06-16, Transcript of Proceedings(ECF 52), <https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.52.0.2.pdf>, Tr. 22-30 — "THE COURT: And what did you find when you went to look up the full Varghese decision? MR. SCHWARTZ: I couldn't find it. THE COURT: And yet you cited it in the brief to me." 심리의 시작과 법정은 ECF 31, 형질수설 문답은 Tr. 30, LoDuca의 말은 Tr. 9(ECF 54 916), 변호인 Minkoff의 말은 Tr. 54. 판사는 심리에서 서면 날짜를 March 15라고 말했지만, 그것은 상대방 답변서의 날짜이고 반대서면은 2023-03-01에 들어갔다(ECF 21). 답변서와 2023-04-11·12 명령, 그 뒤의 경과는 ECF 54 2쪽·917·91913-14 — "unable to locate most of the case law cited" · "the individual Respondents doubled down". ↩
- 29 같은 의견서 1쪽·결론(33-34쪽) — "Technological advances are commonplace and there is nothing inherently improper about using a reliable artificial intelligence tool for assistance. But existing rules impose a gatekeeping role on attorneys to ensure the accuracy of their filings." 2023-05-04 명령의 첫 문장은 ECF 31. ↩
- 30 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.4·§ 3.6, 고친 기준 data/m2_rubric_v2.txt, 정답을 아는 쌍에서 고친 기준이 바꾼 판정은 gpt-6-astra의 이 동점 셋이 정답 쪽으로 간 것뿐이다. 순서를 바꾼 판정과 다시 물은 판정의 원출력은 쌍 A-019(질문 t1c-191) seq data/calls.jsonl seq 1636·1637(gpt-6-astra)·2719(gpt-5.6-sol), 쌍 A-022(질문 t1c-212) seq 1769·1649·2728(세 채점자, 바꾼 순서)·1771(grok-4.7, 다시 물음)이다. 자연 혼합 27쌍에서 오답 쪽을 고른 판정은 바꾼 순서 5개와 다시 물음 2개가 전부이고, 구성 오답 58쌍은 네 조건 모두 정답 쪽이었다. A·B를 답의 정제로 옮기는 대응은 data/m2_pairs.py·data/analyze.py. ↩

고친 자리

trigger

원장의 한 줄은 이렇게 생겼다.

ts	2026-07-13 21:58
family	proxy-green-not-user-visible
trigger	계정 이름 열만 좁혀 달라고 했는데 계정 표 전체가 같이 좁아졌다 (스크린샷 2장)
missed_control_point	PR #80의 완료 조건을 정할 때, '이름 열 20자 이하'를 열을 자르는 렌더 테스트로만 약속했고, 남는 폭이 갈 곳(표는 여전히 전체 폭을 써야 한다)은 조건에 넣지 않았으며, 실제 화면을 눈으로 보지 않고 배포했다
required_hard_artifact	PR #81의 테스트(남는 폭을 게이지가 받아 표가 전체 폭을 쓰는지 확인) + 두 머신에 다시 배포
promotion_artifact	화면을 바꾸는 일은 사용자가 한 말의 절마다 검사 하나를 맞대고 화면 전체를 찍은 증거를 붙여야 끝낼 수 있다는 작업 확인 규칙
status	applied

혼자 일하는 한 실무자의 원장에는 교정마다 상태 칸이 있고, 고친 교정은 '고쳤다
(applied)'로 닫힌다. 그 낱말은 셋 가운데 무엇을 확인했는지 적지 않는다.

이 원장은 내가 개인 에이전트를 부리며 쌓은 것이다. 나는 AI 에이전트에게 일을 맡기고,
틀린 것을 보면 멈춰 세워 바로잡는다. 그때마다 에이전트는 기록 스크립트를 불러 이런
줄을 하나 남긴다. 사용자가 바로잡은 말이 trigger, 그 실패가 드는 무리가 family, 어느 실행
지점에서 막았어야 했는지가 missed_control_point다. 그 자리에서 고친 것은
required_hard_artifact에, 같은 실패가 다시 오지 않게 남긴 장치는 promotion_artifact에
적는다. 마지막 칸 status는 그 정정이 어떻게 닫혔는지다. 위에 옮기지 않은 note 칸에는
같은 자리에서 생긴 다른 일이나 정정이 닫힌 경위가 적히기도 한다. 정정 한 번이 한
줄이라는 것이 원장의 정의다.¹

위의 줄은 원장의 21행이다. trigger 칸의 원문은 이보다 훨씬 거칠다. 여기 옮긴 것은 그
뜻이고, 다른 칸들도 고유한 이름을 빼고 풀어 옮겼다. 원장의 문장은 모두 이렇게
의역해서만 쓴다. 이 줄이 남은 시각은 PR #81이 머지되고 5분 뒤였다.

원장의 첫 여섯 줄은 사람의 정정이 아니다. 2026-07-06 12:57, 앞선 원장에서 실패군마다 한 줄씩 넘겨 온 표식이 한꺼번에 들어왔다. 행의 순서가 시각의 순서와 같지도 않다. 다른 작업 갈래에서 쓴 행이 나중에 합쳐지며 뒤에 붙기도 했다.

status

21행의 마지막 칸에는 applied, 고쳤다는 표시가 있다. 채점자를 재며 가른 세 기준으로 이 낱말을 읽으면, 그것이 무엇을 확인한 표시인지는 칸에 없다. 다른 판정자가 같은 결론을 냈는지를 보는 일치도, 정답과 맞춰 본 정답 정확도도, 고를 수 없는 자리였는지를 가리는 판정 불가도 원장에는 칸이 없다. applied는 기록을 남긴 에이전트가 스스로 적는 값이다. 고쳐졌는지를 따로 확인한 표시가 아니다. status를 담은 값은 applied, rejected, delegated, deleted 넷이다. 고쳤으면 applied, 정정이 맞지 않는다고 보면 rejected, 다른 곳에 맡겼으면 delegated다. 기록 스크립트는 열린 정정이 7일을 넘기면 넷 가운데 하나로 닫으라고 경고한다.²

원장은 2026-07-06부터 10-02까지 214행이다. 넘겨 온 표식 6행을 빼면 실제 정정이 208행이다. 같은 날 trigger가 똑같은 행을 맞대 보면 넷이 겹친다. 08-24의 두 행은 6분 간격으로, 정정 한 번을 두 실패 지점으로 나눠 적었다. 08-27과 08-29의 두 쌍은 같은 사건의 상태가 바뀐 것을 새 행으로 붙인 것이다. 08-28의 두 행은 8시간 간격이라 같은 자리에서 다시 적은 것인지도 분명하지 않다. 원장에는 세션을 적는 칸이 없어서, 같은 자리인지는 시각의 간격으로만 짐작했다. 상태가 바뀐 쌍에서 앞 행은 open이나 delegated였고 뒤 행은 applied였다. 정정이 닫히면 그 줄을 고치는 대신 새 줄을 붙인 것이다. 겹친 넷을 빼고, 같은 사건의 상태는 뒤 행을 따르며 서로 다른 정정은 204건이다.³

주별로 세면 07-06 주부터 13, 27, 15, 4, 9, 22, 11, 47, 9, 8, 4, 11, 24건이다. 가장 많은 08-24 주에는 모드와 범위를 잘못 잡은 실패가 몰렸고, 그 주에는 월요일에 12건, 화요일에 14건이 기록됐다. 가장 적은 07-27 주와 09-14 주에는 4건씩이었다. 이 건수는 그 주에 일한 양에 따라서도 달라질 수 있는데, 일한 양은 확인된 값으로 재지 못했다. 건수가 적은 주를 덜 틀린 주로 읽을 수는 없다.

204건 가운데 applied로 닫힌 것이 188건, 아직 열린 것이 12건, 다른 곳으로 넘긴 것이 3건, 기각된 것이 1건이다. deleted로 닫힌 줄은 없다. 열린 12건은 2026-10-05에 모두

7일 기한을 넘겨 있었고, 가장 오래된 것은 07-17에 열렸다. 그 12건 가운데 7건이 검사는 통과했는데 사용자가 보는 결과가 틀린 무리에 들었다. 사용자의 화면을 확인해야 닫히는 종류라 닫기 어려웠다는 신호일 수도, 우연일 수도 있다. 188이라는 숫자는 에이전트가 188번 고쳤다고 적었다는 뜻이다. 그 가운데 몇 번이 사용자가 보는 결과까지 고친 것인지는 이 원장으로 셀 수 없다.⁴

family

실패군은 여섯이다. 서로 다른 정정 204건을 나누면 이렇다. 빌드 성공이나 200 응답, 몇 군데 찍어 본 확인처럼 보기 쉬운 신호를 유저가 보는 최종 결과 대신 확인한 실패가 42건이다. 원장은 이것을 proxy-green-not-user-visible이라 부른다. 검사는 통과했는데 사용자가 보는 결과는 틀린 실패다. 증거 없이 추론한 실패가 38건, 용어나 출력 형식이 유저의 뜻과 어긋난 실패가 38건, 모드와 범위를 잘못 잡은 실패가 36건이다. 스스로 할 수 있는 일을 질문이나 관문으로 유저에게 넘긴 실패가 30건, 거둬진 신호를 놓친 실패가 20건이다. 가장 많은 무리는 검사 통과 쪽이지만 둘째와는 4건 차이고(겹친 행에서 뒤 행을 남기면 3건), 겹친 행을 빼기 전에는 2건 차이였다. 실패군은 기록하는 에이전트가 그때그때 스스로 고르니, 이 차이가 분류의 흔들림보다 큰지는 이 원장으로 가릴 수 없다.⁵

분류가 처음부터 닫혀 있던 것은 아니다. 2026-06-10부터 07-06까지 4주 가까이 쓴 첫 원장은 정정마다 자유롭게 이름을 붙였다. 108건 가운데 이름이 읽히는 72건에 서로 다른 이름이 43개 붙었고, 그중 35개는 한 번만 쓰였다. 같은 이름이 두 번째 나오면 장치를 만들라는 규칙이 있었지만, 같은 실패가 매번 다른 이름으로 숨는 바람에 그 규칙은 거의 발동하지 않았다. 108건 가운데 그 규칙이 발동했다는 표시가 붙은 줄은 6개였다. 첫 원장에서 그 규칙이 처음 발동한 06-12의 줄도, HTTP 200을 완료로 읽고 smoke 시험 통과를 동작으로 읽은 대리 신호 셋이 잇달아 나온 뒤였다.⁶

2026-07-05의 자체 감사는 기록이 수리를 대신하고 있다고 진단했다. 기계로 검사할 수 있는 실패를 전부 문장으로만 적었고, 혹도 스크립트도 하나 없다는 것이었다. 같은 감사는 보기 쉬운 신호로 끝낸 실패가 첫 원장에서도 가장 큰 몫이었다고 적었고, 정정의 비율을 낮추겠다던 자체 목표가 이뤄지지 않고 있다는 것을 첫머리에 올렸다. 그 근거는 분모 없이 쉰 하루 건수였다. 그 비율을 재는 계기판은 감사 사흘 뒤에 생겼다. 감사 이튿날 시작한 새 원장은 실패군을 여섯으로 단았다. 기록할 때 에이전트는 여섯 가운데 하나를 골라야 했고,

다른 이름은 받아들여지지 않았다. 2026-07-08부터 기록 스크립트는 같은 실패군의 두 번째 정정을 장치 없이는 받지 않았다. 거부할 때 스크립트가 내는 말은 이랬다. 혹이나 스크립트, 테스트, 스킴 계약 같은 장치를 만들고 문장은 쓰지 않으며, 그 장치를 적어 다시 부르라. 넘겨 온 표식 6행이 실패군마다 한 줄씩 미리 들어가 있었으니, 새 원장의 첫 정정부부터 모든 정정이 두 번째로 됐다. 이름을 단자 모든 정정이 여섯 칸 가운데 하나에 들었고, 비로소 셀 수 있게 됐다.⁷ 그 진단과 처방은 같은 에이전트 체계가 두 엔진을 엮갈려 쓰며 내놓은 것이다. 바깥의 검토가 아니었다.⁸

missed_control_point

이 칸은 무엇이 틀렸는지가 아니라 어디서 멈춰야 했는지를 적게 한다. 같은 실패군 안에서도 그 지점은 제각각이다. 검사 통과 무리의 첫 정정은 07-09의 15행이었다. 이 무리와 같은 꼴의 사건 셋이 공개 저장소에 남아 있고, 그 가운데 원장의 행으로 확인되는 것은 첫째뿐이다.

첫째는 21행의 사건이다. 2026-07-13, llmux의 PR #80은 대시보드 계정 표의 이름 열을 20자로 자르는 변경이었다. 옛 동작에서는 실패하도록 짠 렌더 테스트가 붙었다. 본문의 Receipts 절, 곧 증거를 모아 두는 절에는 테스트 두 묶음, 695개와 38개가 통과한 기록이 붙었고, PR의 CI 두 종도 통과했다. PR은 오후 5시 29분(한국 시각)에 머지됐다. 실제 화면에서는 이름 열이 비운 폭이 빈칸으로 남아 표 전체가 줄어 보였다. 4시간 24분 뒤 머지된 PR #81의 본문은 내가 한 말을 옮겨 적었다. 계정 열 너비를 줄여 달라고 했는데 대시보드 계정 표 너비까지 같이 줄였다는 불평이다. #81은 남는 폭을 할당량 게이지 막대들에 나눠 주게 고쳤다. #80의 테스트 이름은 `account_name_column_clips_at_name_col_max`였다. 요청은 이름 열을 20자까지만 쓰라는 것과 지금 너무 넓다는 것뿐이었다. 남는 폭을 이름 열에 주라던 07-09의 지시는 이 요청으로 효력을 잃었지만, 그 폭이 이제 어디로 가야 하는지는 완료 조건에도 테스트에도 없었다. '이름은 20자 이하, 남는 공간은 쓸모 있는 곳으로'라는 의도는 그 불평 뒤에 #81이 적었다. 테스트가 틀린 것이 아니라 시험한 명세가 좁았다. 피해는 되돌리기 쉬운 화면 배치에 그쳤다.⁹

둘째는 같은 저장소에서 이튿날 아침에 났다. 상류 공급자가 429, 곧 너무 많은 요청이라는 응답을 몰아 보낼 때를 다루는 PR #83은 라이브러리 테스트 698개와 통합 테스트 38개,

CI 두 종을 통과하고 머지됐다. 바깥 리뷰가 꼭 고치라고 한 셋 가운데 하나는 고쳤고, 하나는 경계 테스트로 대신했으며(요청 경로 전체를 지나는 통합 테스트는 쓰지 않았다), 하나는 코드로 반박했다. 이번에도 테스트 기록은 본문의 Receipts 절에 붙어 있었다. 머지 37분 뒤 같은 폭주가 다시 났다. 요청 하나가 폴의 계정을 약 8초 동안 13번 훑고 502를 돌려줬다. 상류의 폭주는 20~30초 뒤에 풀렸다. #83의 수정은 Fable 모델 쪽 요청에서만 닿는 코드였고, 07-12부터 15번 이상 되풀이된 Fable 밖 요청의 폭주에는 닿지 않았다. 통과한 테스트의 이름도 `fable_429_burst`로 시작했다. #83이 머지되고 세 시간 남짓 뒤 머지된 PR #85는 운영 기록의 사건을 테스트로 재현해 고정했다. 새로 넣은 통합 테스트는 수정 전 코드에서 4.5ms 만에 실패했다. 고친 코드는 502까지 두드리는 대신 폴이 식기를 기다려 속도를 늦췄고, 약 8초 뒤 200을 돌려줬다. 그 본문의 한 절 제목은 앞선 #72와 #83이 왜 폭주를 막지 못했는가였다. 내가 따로 적어 온 재발 결함 대장도 429 폭주를 막으려던 앞선 수정들이 모두 실제 경로를 빗나갔다고 적는다. 이 사건은 테스트가 손으로 고른 경로였다는 쪽으로도, 운영 기록을 먼저 재현하지 않은 절차의 문제라는 쪽으로도 읽힌다.¹⁰

셋째는 다른 저장소다. 2026-09-17, 에이전트 하네스 `soma-work`의 배포 실행 하나가 1분 남짓 돌고 `success`로 끝났다. 배포 대상 머신 한 대에서는 4시간 전에 시작한 이전 버전의 프로세스가 그대로 돌고 있었다. 같은 날 머지된 PR #226은 그 머신에서의 근본 원인으로, 같은 서비스가 두 관리 영역에 이중으로 등록돼 있었다는 것을 적었다. 그 결과를 가린 사정도 적었다. 프로세스를 멈추다 난 오류는 오류를 버리는 셸 구문에 삼켜졌고, 상태 확인은 서비스 관리자에 남은 옛 프로세스 번호를 실행 중이라고 출력했다. PR은 새 프로세스의 시작 시각이 배포 시작보다 뒤이고 로그에 새 버전 문자열이 찍혀야만 통과하는 확인을 넣고, 거기에 `verify-restart`라는 이름을 붙였다. 제목은 이렇다. "정직한 정지와 재시작 확인 — 남은 프로세스 번호 위의 초록 배포는 더 없다." 그날 저녁 7시 10분(한국 시각), 머지 뒤 그 머신으로 간 첫 배포는 `red`, 실패로 끝났다. 새 프로세스가 뜨지 못했고, 이번에는 그 실패가 숨지 않았다. 이 사건은 언어 모델의 출력이 아니라 배포 스크립트의 일이다. 검사를 통과했는데 사용자가 보는 결과가 틀린 일은 모델 없이도 같은 꼴로 일어났다.¹¹

세 사건 모두에 통과한 검사가 있었다. 렌더 테스트는 이름 열이 잘리는지만 봤고, 429 테스트는 Fable 쪽 경로만 지났고, 배포의 상태 확인은 프로세스 번호가 살아 있는지만 물었다. 통과한 검사는 사용자가 볼 결과의 한 조각을 시험했다.

promotion_artifact

규칙이 요구한 장치는 쌓였다. 실제 정정 208행 가운데 promotion_artifact 칸에 실행되는 장치를 적은 행이 108행이다. 혹 22, 테스트 14, CI 3, 스크립트 69. 스크립트를 적은 69행 가운데 54행의 장치는 이름이나 설명이 gate, lint, check, verify, receipt, scan 같은 낱말이거나 실패하면 멈춘다고 적은 검사기였다. 장치들은 실제로 만들어졌다. 108행 가운데 100행의 장치가 저장소 이력이나 디스크에서 확인되고, 혹 22행은 서로 다른 혹 파일 9개로 모두 확인된다. 한 파일에 10행이 덧붙은 것도 있다. 종류마다 남은 정도는 달랐다. 테스트는 14행 가운데 13행이 확인되고 12행의 장치가 남아 있으며, CI는 3행이 모두 확인되고 2행의 장치가, 스크립트는 69행 가운데 62행이 확인되고 59행의 장치가 남아 있다. 2026-10-05 현재 그 저장소에 남은 혹은 없다. 종류 전체가 사라진 것은 에이전트의 행동을 검사하던 혹뿐이었다.¹²

이 108행은 실패를 겪은 뒤 저절로 나온 수가 아니다. 넘겨 온 표식 때문에 모든 정정이 두 번째로 켜고, 장치 없이는 기록을 받지 않는 규칙이 그 칸을 채우게 했다. 같은 행의 required_hard_artifact, 그 자리에서 고친 것의 칸에 실행되는 장치가 적힌 행은 24행뿐이었고, 143행은 문장이었다. 겹친 4행 가운데 3행이 이 108행에 들어 있어, 겹친 행을 빼면 장치를 적은 행은 105행이다.

그 장치들이 생긴 뒤 같은 실패가 다시 기록됐는지는 원장으로 따라갈 수 있다. 아래 표의 첫 장치는 원장이 있는 저장소의 커밋으로 확인되는 첫 장치다. 교정 행마다 그 행이 적은 장치 경로를 추가하거나 고친 커밋 가운데 행의 시각 전후 48시간 안에서 가장 가까운 것을 그 정정에 응답한 커밋으로 보고, 실패군마다 가장 이른 것을 첫 장치로 뒀다. 그 전과 그 뒤는 그 커밋의 앞뒤로 기록된 같은 실패군의 교정 건수이고, 둘을 더하면 그 실패군의 전체 건수다. 그 장치의 이후는 2026-10-02까지 그 장치가 고쳐졌는지와 저장소에 남았는지다.¹³

실패군	첫 교정	첫 장치	그 전 / 그 뒤	그 장치의 이후
검사 통과, 사용자 결과는 틀림	07-09	08-05 혹 신설	13 / 29	고친 적 없음, 10-01 커밋으로 삭제
증거 없이 추론	07-07	07-08 스크립트 신설	1 / 37	고친 적 없음, 남아 있음

실패군	첫 교정	첫 장치	그 전 / 그 뒤	그 장치의 이후
모드·범위 오판	07-08	07-29 혹 신설	10 / 26	고친 적 없음, 10-01 커밋으로 삭제
용어·형식 어긋남	07-08	07-08 혹 신설	1 / 37	고친 적 없음, 10-01 커밋으로 삭제
할 일을 유저에게 넘김	07-10	08-25 기존 혹 수정	14 / 16	여섯 번 고침, 10-02 커밋으로 삭제
거듭된 신호를 놓침	07-07	07-09 스크립트 수정(같은 날 신설)	2 / 18	고친 적 없음, 남아 있음

여섯 실패군 모두 첫 장치가 생긴 뒤에도 같은 실패군의 교정이 이 원장에 다시 기록됐다. 적게는 16건, 많게는 37건이다. 원장이 있는 저장소에서 확인되는 검사 통과 무리의 첫 장치는 08-05에 생긴 혹이었고, 그 뒤 주별 건수는 0, 4, 0, 8, 2, 1, 1, 5, 8로 이어졌다. 이 무리는 첫 정정인 07-09의 15행부터 장치를 적었고, 모드·범위 오판 무리와 할 일을 유저에게 넘기는 무리에도 07-15부터 표의 첫 장치보다 앞서 장치를 적은 행이 있었다. 그 행들은 원장이 있는 저장소에서 행의 시각 전후 48시간 안에 그 장치를 추가하거나 고친 커밋이 보이지 않아 표에 들지 않았다. 표의 '그 전'은 장치가 없던 동안의 건수가 아니다. 정정에 응답해 장치를 고친 커밋은 실패군마다 5~20개였고, 마지막 것들은 09-30~10-02에 있다. 할 일을 유저에게 넘기는 무리의 첫 장치는 새 혹이 아니라 이미 있던 혹을 고친 것이었고, 그 혹은 10-01까지 여섯 번 더 고쳐졌다. 그 뒤로도 같은 무리의 정정은 16건이 기록됐다. 장치가 막아 낸 사건은 이 원장으로 셀 수 없다. 행은 정정이 일어난 뒤에만 생긴다. 장치가 없었다면 몇 건이 더 기록됐을지도 이 원장은 말하지 않는다. 이 원장에서 확인하지 못한 것도 있다. 2026-10-02 뒤의 일, 에이전트가 기록하지 않은 정정, 다른 저장소나 다른 기계의 장치, 그리고 실패군 분류가 맞았는지. 장치를 적은 105행 가운데 37행은 행의 시각 전후 48시간 안에 그 장치를 고친 커밋이 원장이 있는 저장소에서 보이지 않았다. 장치가 다른 저장소에 있었거나, 커밋되지 않았거나, 고친 시각이 48시간 밖이었던 경우다.¹⁴

장치를 요구하던 규칙은 2026-10-01에 지워졌다. 그날 기록 스크립트의 머리 주석은 여섯 실패군을 집계용 분류라고 부르고, 같은 실패군이 거듭돼도 기록은 거부하지 않는다고 고쳐 적었다. 같은 실패군의 두 번째 정정이라도 이제 장치 없이 기록할 수 있었고, 그 뒤 기록된

6행 가운데 3행은 promotion_artifact 칸을 비운 채 남았다. 표의 혹들도 그 무렵의 정리에서 지워졌다. 스크립트 둘은 저장소에 남았다.¹⁵

note

이 원장으로 그 석 달 동안 덜 틀리게 됐는지를 말하려면 분모가 있어야 한다. 같은 주에 얼마나 일했는지도. 후보는 둘이었다. 하나는 내 계기판이 쓰던 분모, 그 주의 작업 로그 항목 수다. 날마다 한 일을 적는 로그의 머리줄을 센 것이다. 다른 하나는 그 주에 시작된 세션 기록의 수다. Claude Code는 대화마다 기록 파일을 남기는데, 원장이 있는 작업 폴더에서 시작된 것만 세고 같은 시각에 되풀이되는 정기 실행 109개는 뺐다. 세션 기록 쪽은 Claude Code가 기본 설정으로 30일 넘은 기록을 지우는 바람에 앞쪽이 잘려 있었다. 그 기간을 바꾼 설정은 없었다. 이 기계에 남은 세션 기록 714개 가운데 가장 오래된 수정 시각은 09-02 22:34였다. 08-31 주까지는 주마다 0~23개만 남았다. 30일 넘은 기록만 지워졌다고 보면, 빠짐없이 남은 주는 09-07부터 4주뿐이었다. 13주 가운데 9주가 잘려 나간 셈이다.

세션 기록은 이 기계의 것만이고, 다른 작업 폴더에서 시작된 310개와 다른 기계의 기록은 빠졌다. 작업 로그 쪽은 13주가 다 있지만, 측정한 날 저장소에 합쳐져 있던 로그만 센 것이고 그것이 일의 양을 따라가는지는 확인되지 않았다. 어느 쪽도 유저가 에이전트에게 말을 건넌 횟수, 곧 정정이 일어날 수 있었던 기회는 아니다. 두 분모가 겹치는 4주에서 세션 수는 47, 85, 24, 85개였고, 같은 주의 로그 항목 수는 17, 17, 50, 108개였다. 상관은 0.27에 그쳤다. 로그 항목 수를 세션 수로 나눈 값은 주마다 0.20에서 2.08까지, 열 배 넘게 흔들렸다. 같은 4주에서 주별 기울기의 추정값은 분모에 따라 부호가 반대였고, 어느 쪽도 0과 구별되지 않았다. 일별로 보면 정정 건수는 세션 수보다 로그 항목 수와 더 함께 움직였다. 작업 로그도 정정의 행도 같은 에이전트가 쓰니, 그것은 로그가 일의 양을 잘 재서일 수도 있고 같은 기록 습관 때문일 수도 있다. 이 자료로는 가를 수 없다. 감사의 판정은 둘 다 맞지 않는다는 것이었다. 그래서 여기 적은 것은 건수와 관측 범위뿐이다. 장치를 적은 행이 108행 쌓이는 동안 내 계기판은 하락을 보이지 않았고, 그 분모는 확인되지 않았다.¹⁶

원장이 보여 주는 범위는 좁다. 정정이 원장에 남으려면 사람이 에이전트를 멈춰 세워야 하고, 에이전트가 기록 스크립트를 불러야 한다. 둘 가운데 하나라도 빠지면 그 실패는

원장에 없다. 실패군은 기록하는 에이전트가 스스로 고른다. 그 실패를 낸 에이전트가 어느 모델로 돌아났는지를 적는 칸은 없다. 한 사람이 한 시스템을 석 달 가까이 부린 기록이다. 원장은 시스템에 장치가 붙은 자리를 보여 주고, 그 사이에 사람과 에이전트가 덜 틀리게 됐는지는 보여 주지 않는다.

ts

관측이 끝나는 2026-10-02, 원장의 마지막 행은 이렇다.¹⁷

ts	2026-10-02 18:21
family	ignored-repeated-signal
trigger	쉴 브라우저의 이름은 이미 쫓다는 한마디
missed_control_point	유저가 쉴 브라우저를 이름으로 지정한 직후, 연결된 브라우저 목록에서 그 이름을 찾아 바로 골랐어야 했는데, 모든 브라우저에 연결 확인 창을 띄워 클릭을 유저에게 넘겼다
required_hard_artifact	이름을 받은 브라우저는 목록에서 찾아 곧바로 고른다. 확인 창은 유저가 그 방식을 고른 경우에만 띄운다
promotion_artifact	
status	applied

주

- 1 원장 21행(2026-07-13 21:58, 한국 시각). 원장은 비공개 저장소에 있어 문장을 의역했다. 이 원장은 [data/a1-audit-results.md](https://github.com/2lab-ai/llmux/pull/80)가 집계한 원장(214행)이고, 칸의 구성은 원장 규격(비공개)을 따른다. 이 행이 가리키는 공개 PR은 <https://github.com/2lab-ai/llmux/pull/80> 과 <https://github.com/2lab-ai/llmux/pull/81> 이다. ↩
- 2 저자 집계, [data/a1-audit-results.md](#) § 5의 끝(applied 집계). status 칸을 채우는 것은 기록하는 에이전트다. status 값의 목록과 7일 경고는 기록 스크립트(비공개)의 것이다. 값의 뜻은 규격에 따로 적혀 있지 않아 낱말 뜻대로 풀었다. ↩
- 3 저자 집계, [data/a1-audit-results.md](#) § 1(집친 4행과 묶음별 간격)·§ 2 표(주별 건수 N). 08-24의 12건은 § 1, 08-25의 14건은 [data/a1_ledger_stats.py](#)로 센 저자 집계다(그날 집친 행은 없다). 관측 범위는 2026-07-06~10-02, 한 사람의 한 시스템이다. ↩
- 4 저자 집계, [data/a1-audit-results.md](#) § 5의 끝 — 집친 행을 뺀 204건의 상태(같은 사건은 뒤 행의 상태를 따름). 열린 정정의 실패군별 수(검사 통과 무리 7건)는 집친 행을 빼기 전의 저자 집계([data/a1_ledger_stats.py](#))이고, 빠진 열린 행 하나는 다른 실패군이라 7건은 그대로다. ↩
- 5 저자 집계, [data/a1-audit-results.md](#) § 1 — 집친 행을 뺀 실패군별 건수(앞 행 유지). 원장의 실패군 이름은 proxy-green-not-user-visible · infer-without-evidence · jargon-or-output-mismatch · wrong-routing-or-scope · handoff-or-gate-instead-of-doing · ignored-repeated-signal이다. ↩
- 6 저자 집계(집계 방법 [data/a1_ledger_stats.py](#), 입력 원장은 비공개). 이름 계수는 문자열 정규식에 기댄 근사이고, 첫 원장(2026-06-10~07-06)은 이 원장의 관측 범위 밖이다. ↩
- 7 실패를 읽어 범주로 묶는 일을 LLM 평가 실무는 늦어도 2025-06부터 개방 코딩·축 코딩이라 불렀다. David G  rouville-Farrell, 2025-06-21, "LLM Evals Lesson 2 Error Analysis"(Hamel Husain-Shreya Shankar 강의 2장의 수강 노트), thingsithinkithink, <https://thingsithinkithink.blog/posts/2025/06-21-llm-evals-lesson-2-error-analysis/> — "This introduces the open coding technique, which draws from qualitative research methodology." · 절 제목 "Cluster Failure Modes (Axial Coding)". Hamel Husain-Shreya Shankar, 2025-07-01판, "Frequently Asked Questions (And Answers) About AI Evals"(2025-07-03 보존본), <https://web.archive.org/web/20250703120004/https://hamel.dev/blog/posts/evals-faq/> — "Axial Coding: Categorize the open-ended notes into a "failure taxonomy."" 근거 이론의 원래 정의에서 축 코딩은 범주를 하위 범주와 관계 짓고 그 관계를 자료로 시험하는 일이었다. Juliet Corbin-Anselm Strauss, 1990-12, "Grounded Theory Research: Procedures, Canons and Evaluative Criteria", Zeitschrift f  r Soziologie 19(6):418-427, p.423 — "In axial coding categories are related to their subcategories, and these relationships tested against data." Corbin 단독의 2008년판은 두 단계를 따로 정의하지 않는다고 Stol-Ralph-Fitzgerald(2016-05, ICSE 2016, pp.120-131)가 정리했다. ↩
- 8 저자의 2026-07-05 자체 감사, 원장 규격, 기록 스크립트의 거부 문구(모두 비공개)를 의역했다. 기록을 받지 않는 규격은 2026-07-08부터 2026-10-01까지 있었다. 계기판 스크립트가 처음 커밋된 것은 2026-07-08이다(저자 저장소 이력, 비공개). 감사가 첫머리에 올린 목표의 계기판은 뒤의 본문 감사에서 본문이 확인되지 않았다([data/a1-audit-results.md](#) § 3). ↩
- 9 저자, 2026-07-13, llmux PR #80 "fix(tui): cap the accounts name column at 20 chars"(머지 08:29:58Z), [h](https://github.com/2lab-ai/llmux/pull/80) [tps://github.com/2lab-ai/llmux/pull/80](https://github.com/2lab-ai/llmux/pull/80) · PR #81 "fix(tui): pour leftover accounts-table width into the quota gauge bars"(머지 12:53:38Z), <https://github.com/2lab-ai/llmux/pull/81> — "the freed width became dead space and the whole table visually shrank." PR #80-#81 본문이 옳긴 내 말은 의역했다. ↩

- 10 저자, 2026-07-13, llmux PR #83 "fix(proxy): 429 grace-park budget — ride out bursts + pre-emptive hard cap"(머지 22:24:06Z), <https://github.com/2lab-ai/llmux/pull/83> · PR #85 "fix(proxy): pace a non-Fable 429 sweep on the pool cooldown instead of hammering to 502"(머지 2026-07-14 01:42:25Z), <https://github.com/2lab-ai/llmux/pull/85> — "Root cause — why #72/#83 didn't stop it". 운영 환경은 저자 개인 프록시의 작은 규모다. 재발 결합 대상은 저자의 비공개 문서다. ↩
- 11 저자, 2026-09-17, soma-work PR #226 "fix(deploy): honest stop + restart verification (no more green deploys on a stale PID)"(머지 06:46:20Z), <https://github.com/2lab-ai/soma-work/pull/226> · 배포 실행 <https://github.com/2lab-ai/soma-work/actions/runs/35184945142> (success, 05:14-05:15Z) · <https://github.com/2lab-ai/soma-work/actions/runs/35209063075> (failure, 10:10Z). PR 본문의 머신 이름은 옳기지 않았다. ↩
- 12 저자 집계(집계 방법 `data/a1_ledger_stats.py`, 입력 원장은 비공개) — 208행의 `promotion_artifact`를 정해진 정규식 순서로 나눈 값과, 장치가 저장소 이력·디스크에 있는지 2026-10-05에 대조한 값. 겹친 행을 뺀 105행 (혹 21 · 테스트 13 · CI 3 · 스크립트 68)은 `data/a1-audit-results.md` § 1. 분류가 경로·확장자·날말에 기대므로 기준을 바꾸면 수가 달라진다. 다른 저장소의 장치는 로컬 체크아웃 기준이라 적게 썼을 수 있다. ↩
- 13 저자 집계, `data/a1-audit-results.md` § 5(규칙과 표). 가둬진 신호 줄의 '같은 날 신실'은 저자 저장소 이력 (비공개)이다. ↩
- 14 같은 절. 표의 첫 장치보다 앞서 장치를 적은 행(검사 통과 15:32~63행, 모드·범위 오픈 33~39행, 할 일을 유저에게 넘김 34~47~88행)은 `data/a1_ledger_stats.py`로 쓴 저자 집계다. 장치와 실패군의 연결도 에이전트의 자기 기록이다. ↩
- 15 저자 집계, `data/a1-audit-results.md` § 5(그 장치의 이후). 규칙을 지운 뒤의 표본은 6행·이틀이라 규칙이 있을 때와 견줄 수 없다. ↩
- 16 저자 집계, `data/a1-audit-results.md` § 2(본모 돌, 보존 절단, 주별 표, 상관과 기울기)·§ 3(판정)·한계(보존 절단의 가정). 세션 기록의 보존 기간은 Claude Code 문서 <https://code.claude.com/docs/en/claude-director>의 `cleanupPeriodDays`(기본 30일). 같은 4주의 상관은 정기 실행을 뺀 세션 기준 $r + 0.27$ 이고, 주별 기울기는 작업 로그 기준 -7.60% /주(주를 복원추출한 부트스트랩 95% 구간 $-23.53 \sim +0.22$), 같은 세션 기준 $+7.48\%$ /주($-17.60 \sim +41.13$)였다. 두 구간 모두 0을 포함한다. 일별(09-03~10-02, $n=30$) 상관은 정정 건수와 로그 항목 수 $\rho + 0.71$, 정정 건수와 세션 수 $\rho + 0.29$ 다. ↩
- 17 원장 214행(2026-10-02 18:21, 한국 시각). 의역했고, 유저가 지정한 브라우저의 이름은 옳기지 않았다. 이 행의 `promotion_artifact` 칸은 비어 있다. ↩

삼십 퍼센트

문다

"What is the answer?"

2023년 가을 학기, 튀르키예의 한 고등학교에서 묻는 대로 답하는 GPT-4를 받은 반의 학생들이 가장 많이 보낸 메시지가 이 한 줄이었다. 답이 뭐냐는 것이다. 9학년부터 11학년까지 약 50개 반, 1,000명 가까운 학생이 90분짜리 수학 회기 네 번을 치렀고, 네 회기를 합치면 그 학기 수학 진도의 15% 안팎이었다. 조건은 반 단위로 무작위로 갈렸다. 한 무리의 반은 ChatGPT처럼 묻는 대로 답하는 GPT-4를 받았다. 연구자들은 이것을 GPT Base라고 불렀다. 다른 무리가 받은 GPT Tutor는 같은 GPT-4에 문제마다 정답과 혼한 오답, 교사가 짠 힌트를 넣고 답을 곧바로 주지 말라고 지시한 판이었다. 나머지 반은 AI 없이 연습했다. 교사와 직원이 교실을 지키며 학생들이 다른 앱을 쓰는지 살폈다.¹

연구자들은 학생들이 가장 많이 쓴 그 한 줄로 GPT Base를 직접 시험했다. 연습 문제 57개마다 "What is the answer?"를 열 번씩 넣었더니 정답은 평균 51%였다. 42%에는 논리 오류가, 8%에는 산술 오류가 있었다.²

연습 문제 점수는 두 AI 무리에서 모두 대조군보다 높았다. GPT Base 반은 48%, GPT Tutor 반은 127% 높은 점수를 냈다. AI 없이 치른 시험은 다르게 나왔다. GPT Base 반은 대조군보다 17% 낮았고, GPT Tutor 반은 대조군과 통계적으로 다르지 않았다. 스스로 어땠다고 보는지 묻자 GPT Base 반 학생들은 시험을 더 못 봤다거나 덜 배웠다고 여기지 않았다. GPT Tutor 반 학생들은 시험에서 대조군보다 낮지 않았는데도 더 잘 봤다고 여겼다. 연구자들은 학생들의 자기 평가가 "지나치게 낙관적(overly optimistic)"이라고 썼다.³

도움이 있는 동안 켄 점수에서 GPT Base 반은 대조군 위에 있었고, 도움을 곁은 시험에서는 아래에 있었다. 학생들의 자기 평가에는 그 아래쪽이 나타나지 않았다. 이 결과는 튀르키예의 한 학교에서 수학 한 과목으로, 2023년 가을의 GPT-4를 써서 켄 단기

결과다. 연구자들도 다른 튜터 설계와 다른 환경으로 넓히려면 연구가 더 필요하다고 적었다.

그 연구의 설계에서 두 점수는 맑은 일이 달랐다. 연습 점수는 도움이 곁에 있는 동안의 기록이었고, 배움은 같은 회기 끝에 도움을 걷은 시험에서 잤다.⁴

걷어 내다

204건의 원장은 시스템에 장치가 붙은 자리를 보여 줬다. 그 석 달 동안 사람이 덜 틀리게 됐는지는 보여 주지 않았다. 그것을 말하려면 같은 주에 얼마나 일했는지가 분모로 있어야 했는데, 후보 둘 다 쓸 수 없었다.⁵ 사람이 무엇을 배웠는지는 기록 밖에 있었다. 튀르키예의 그 학교에서 연구자들은 배움을 기록 대신, AI를 걷어 낸 시험으로 잤다.

Robert Bjork는 1994년 사람의 훈련을 다룬 글에서 훈련 중의 수행과 훈련 뒤의 수행을 갈라 봤다. 훈련 중 습득을 빠르게 하는 조작은 훈련 뒤의 장기 수행을 받쳐 주지 못할 수 있고, 훈련 중에 학습자를 힘들게 하는 것처럼 보이는 조작이 훈련 뒤의 수행을 높일 수 있다는 것이다. 그가 든 조작은 연습 조건 바꾸기, 섞어서 연습하기, 간격 두기, 시험 보기, 피드백 줄이기였고, 그는 이것들을 '바람직한 어려움(desirable difficulties)'이라고 불렀다. 무게는 '바람직한'에 실린다. 어떤 조합이 나은지는 과제마다 다르다고 그는 단서를 달았다.⁶

그가 요약한 Bahrick의 1979년 실험에서는 스페인어 낱말 50개를 0일, 1일, 30일 간격의 회기로 익혔다. 마지막 회기 30일 뒤의 회상은 0일 간격이 33%, 1일 간격이 64%, 30일 간격이 72%였다. 비행기에서 구명조끼 시연을 수없이 보는 것보다 한 번 직접 입어 보는 편이 낫다는 예도 같은 글에 있다.

그 학교의 두 점수는 이 구분을 따라 놓여 있었다. 연습 점수는 훈련 중의 수행 쪽에 있었고, 같은 회기 끝에 도움을 걷고 잤 시험 점수는 그 반대쪽에 있었다. 연구자들은 자신들이 본 것이 장기 학습이 아니라 시험 점수로 추정된 단기 학습이라고 적었다.⁷

안심하다

학습자 자신도 훈련 중의 수행에 속는다고 Bjork는 썼다. 수행이 빨리 좋아지면 배움이 거의 없어도 안심하고, 해매며 틀리면 배움이 상당해도 괴롭다. 그래서 학습자가 덜 효과적인 훈련을 더 좋아하게 될 수 있다.⁸ 학습자가 자기 배움을 재는 잣대를 그는 '믿을 수 없는 지표'라고 불렀다. 지금 이 순간 정답이나 절차에 얼마나 쉽게 닿는지다.⁹

Asher Koriat와 Bjork는 2005년 낱말 쌍 실험으로 그 착각이 생기는 자리 하나를 보였다. 공부하면서 내리는 '나중에 기억하겠다'는 판단은 문제와 답이 함께 눈앞에 있을 때 내려지고 시험에는 그 답이 없어서, 그 차이를 덜어 내지 못하면 공부하는 동안 '능력 착각'이 생길 수 있다.¹⁰ 이 기제가 그 교실의 낙관을 만들었는지는 그 교실의 연구가 재치 않았다.

맞히다

1972년 7월부터 1976년 6월까지, 미국 기상청 시카고 예보소의 예보관들은 강수확률 예보 17,514건을 냈다. 비나 눈이 올 확률을 숫자로 적는 예보다. 그 4년 동안 '30%'라고 적은 것은 1,574번이었다. 그 가운데 449번, 28.5%에 측정할 만한 강수(0.01인치 이상)가 있었다.¹¹

Allan Murphy와 Robert Winkler는 1977년 이 기록을 분석했다. 예보한 확률과 실제로 강수가 있었던 빈도는 전체적으로 가까웠다. 예보관들은 실제보다 조금 높게 예보하는 경향을 보였고, 높은 확률은 드물게만 썼다.

채점하다

확률 예보를 실제 결과와 견줘 점수를 매기는 방법은 1950년 미국 기상국의 Glenn Brier가 내놓았다. 지금 Brier 점수라고 부르는 것이다. 그 논문에서 그는 만족스러운 검증의 필수 기준 하나를 이렇게 적었다. 검증 방식은 예보관에게 바람직하지 않은 쪽으로 영향을 주어서는 안 된다.¹² 채점 방식에 맞춰 자기 판단과 다른 숫자를 적는 일을 그는 'hedging'이라고 불렀다.

피드백이 많을수록 낫다는 기대와 어긋나는 대목이 Bjork의 같은 1994년 글에 있다. 그는 과제별로 묶은 연습, 계속 주는 피드백, 고정된 연습 조건 같은 조작이 훈련 중의 수행을 인위적으로 떠받치는 목발처럼 작동하고, 훈련 뒤에 그 목발이 없으면 수행이 무너진다고 썼다.¹³

어긋나다

Robin Hogarth는 2001년 책 『Educating Intuition』에서 학습 환경을 '친절한(kind)' 환경과 '사악한(wicked)' 환경으로 나눴다. 2015년 Tomás Lejarraga, Emre Soyer와 함께 쓴 논문은 그 정의를 이렇게 정리했다. 사악한 환경은 행동의 결과나 관찰로 오는 피드백이 빈약하거나 오도하거나 아예 없는 상황이다. 친절한 환경에서는 피드백이 결과를 알맞은 행동이나 판단에 곧바로 잇고, 그 피드백이 정확하고 풍부하다. 논문은 친절한 환경을 정확한 직관의 필요조건으로 두고, 충분조건으로 두지는 않는다.¹⁴ 이 'wicked'는 소프트웨어 공학 글에 자주 나오는 Rittel과 Webber의 'wicked problem'(1973)과 다른 개념이다.

그 책이 '사악한'을 소개하며 든 이야기가 있다. 20세기 초 뉴욕의 한 병원 의사는 환자의 혀를 손으로 만져 보고 장티푸스를 예견했고, 예견은 맞았다. 그 손이 병을 옮기고 있었다. 이 의사의 이야기를 남긴 Lewis Thomas는 그가 "손만으로 장티푸스 메리보다 더 효과적인 보균자였다"고 썼다.¹⁵ 결과는 그의 판단을 확인해 줬고, 그 확인이 그를 오도했다.

2015년 논문은 이 구분을 두 설정 사이의 맞음과 어긋남으로 다시 썼다. 배울 때 보는 학습 설정(L)과, 배운 것을 쓸 때 만나는 목표 설정(T)이다. 두 설정의 정보가 가깝게 맞으면 친절한 환경이고, 어긋나면 사악한 환경이다.¹⁶ 무작위로 생긴 어긋남은 이 틀에서 친절한 쪽에 든다. 예측력을 크게 떨어뜨려도 그렇다. 사악함은 체계적인 어긋남에서 나온다.¹⁷

이 틀로 읽으면, 그 학교의 GPT Base 반에서 학습 설정에는 답을 주는 GPT-4가 있었고 목표 설정인 시험에는 없었다. 그 어긋남은 무작위가 아니었다. 연습하는 동안에는 같은 도움이 늘 곁에 있었고, 시험에는 없었다. Bjork가 목발이라 부른 계속 주는 피드백도 같은 모양으로 읽힌다. 그 피드백은 학습 설정에 있었고 목표 설정에는 없었다. 시카고의 예보관들은 두 설정이 거의 겹치는 자리에 있었다. Hogarth가 2003년 에세이에서 든 기상예보관처럼, 예보마다 곧 결과가 나왔다.¹⁸

믿다

직관을 언제 믿을 수 있는지를 다룬 2009년 논문에서 Daniel Kahneman과 Gary Klein은, 숙련된 직관이 자라려면 타당성이 충분히 높은 환경과 그 기술을 연습할 적절한 기회, 곧 오래 이어지는 연습과 빠르고 모호하지 않은 피드백이 함께 있어야 한다고 썼다.¹⁹

옮겨 가다

표지의 첫 뜻(틀려도 괜찮은 연습)을 그 교실에 대 보면, 2023년 가을 튀르키예의 한 고등학교에서 약 1,000명이 참여한 그 실험에서 답을 주는 GPT-4와 연습한 학생들의 연습 점수는 올랐다(+48%). 그 연습 점수는 시험 점수와 함께 최종 성적에 들어갔다.

그 점수는 도움 없이 치른 시험으로 옮겨 가지 않았다. 그 시험에서 GPT Base 받은 대조군보다 17% 낮았다. 학생들의 자기 평가는 그 차이를 반영하지 못했다. 연구자들은 해의 원인을 학생들이 GPT-4를 '목발(crutch)'로 삼아 답을 받아 베낀 데서 찾았다. Bjork가 계속 주는 피드백 같은 조작에 붙었던 그 낱말이다. 연구자들이 든 근거 가운데 하나는 GPT Base의 오류가 연습 점수는 깎았지만 시험 점수와는 유의한 관계를 보이지 않았다는 분석이었다. GPT Base 반 학생들은 가장 흔히 답만 물었다고도 그들은 적었다.²⁰ 열 번의 시연 화면에 오답이 하나도 없던 그 앱은 240개에서 17.1%를 틀렸다.²¹ 그 열 번은 오답이 없는 연습, 곧 틀린 것이 화면이나 점수에 보이지 않는 연습이었다. 그 앱의 시연 화면도, 그 교실의 연습 점수도, 그 표시를 만든 조건 밖에서 무엇을 할 수 있는지는 말하지 않았다.

그 앱은 claude-haiku-4-5로 돌린 한 과업이고, 그 교실은 튀르키예의 한 학교다. 둘을 나란히 놓는 것은 구조가 닮았다는 유비다. 친절한 환경과 사악한 환경의 틀을 LLM 위에서 제품을 짓는 일이나 그 일을 배우는 사람에게 직접 시험한 실증 연구는 2026-10-05까지 찾은 범위에서 나오지 않았다. 시스템 설계에 이 틀을 빌려 쓴 엔지니어의 글은 있다.²² 실험 증거는 학습자 쪽에 있다.

닮은 결과는 다른 집단에서도 나왔다. 2026년 1월 Anthropic의 연구자들이 낸 무작위 실험에서는 비동기 파이썬 라이브러리 Trio를 처음 쓰는 개발자 52명(크라우드워커, 조건당 26명)이 과제를 마친 뒤 AI 없이 퀴즈를 봤고, AI 보조 집단의 점수가 17%p 낮았다(저자들의 글로는 평균 50% 대 67%). 이 표본에서는 자기 보고 학습량도 AI를 쓰지 않은

집단이 더 높았다. 체감이 측정과 같은 쪽을 가리킨 표본이다.²³ 2022년의 코드 모델과 함께 보안 관련 프로그래밍 과제 다섯 개를 푼 47명(AI 보조 33명, 대조 14명, 3분의 2가 학부생)의 실험에서는 AI 보조 집단이 네 과제에서 불안정한 답을 더 자주 냈고, 평균적으로 자기 답이 더 안전하다고 믿었다. 한 참가자는 "나중에 테스트하겠지만 지금은 내 AI를 믿겠다"고 적었다.²⁴

답을 곧바로 주지 않게 만든 GPT Tutor 받은 연습 점수가 대조군보다 127% 높았지만, 시험에서는 대조군과 통계적으로 다르지 않았다(점 추정 -0.004 , $0\sim 1$ 로 정규화된 점수에서 대조군 평균은 0.321). 그 반 학생들은 메시지를 더 많이 보냈고, 스스로 풀어 본 시도와 도움 요청도 더 많았다. 연구자들은 해가 사실상 사라졌지만 긍정적인 효과는 여전히 보이지 않았다고 썼다.²⁵

보정하다

시카고의 예보관들이 '30%'라고 적은 1,574번 가운데 449번 강수가 있었다. 4년 동안 17,514건의 예보가 결과와 하나씩 맞대어졌다. 그렇게 검증된 예보에서 이런 보정이 관측됐다. Murphy와 Winkler는 예보관 대부분이 이런 예보에 경험이 많고, 객관적 방법으로 낸 참고 확률(가이던스)도 받아 볼 수 있다고 적었다.²⁶ 그 검증이 본 것은 측정할 만한 강수가 실제로 있었는지였다. 검증이 그 보정을 만들었는지는 이 기록이 가르치 않는다.

Brier가 검증에 건 기준은 이 관측과 다른 층에 있다. 그는 검증 방식이 예보관을 자기 판단과 다른 숫자 쪽으로 이끌어서는 안 된다고 적었다. 예보가 결과와 맞았는지가 아니라, 검증을 어떻게 짤지에 대한 기준이다.

검증이 목표와 어긋난 쪽을 보상하면, 그 통과는 목표의 증거가 아니다. 그 학교에서 목표는 도움 없이 푸는 실력이었다. GPT Base 반의 연습 채점은 GPT-4가 곁에 있는 동안 낸 답에 점수를 줬고, 연습 점수가 오른 것은 그 채점을 통과했다는 기록이었다. 목표에 닿았는지는 도움을 걷은 시험에서 따로 잰다. 장티푸스를 예견한 의사가 결과에서 받은 확인도 통과였다. 그 통과를 만든 것은 병을 옮기던 그의 손이었다.

종이

학생들이 연습 답안을 종이에 적어 내면, 교사가 반 전체와 정답을 짧게 훑었다. 같은 회기 끝에, 학생들은 AI 없이 같은 범위의 시험을 치렀다.²⁷

주

- 1 Hamsa Bastani·Osbert Bastani·Alp Sungur·Haosen Ge·Özge Kabakcı·Rei Mariman, 2025-06-25, "Generative AI without guardrails can harm learning: Evidence from high school mathematics", PNAS 122(26):e2422633122, <https://doi.org/10.1073/pnas.2422633122> (전문 <https://pmc.ncbi.nlm.nih.gov/articles/PMC12232635/>), 연구 설계 — "We conducted four 90-min sessions for about fifty 9th, 10th, and 11th-grade classes, comprising nearly 1,000 students." 2025-08-20의 정정(PNAS 122(34))은 소속 표기를 고친 것으로 결과와 무관하다. ↩
- 2 같은 논문, 기제 분석 — "For each of the 57 total practice problems, we ask GPT Base for the answer ten times" · "GPT Base gives a correct answer only 51% of the time on average". ↩
- 3 같은 논문, 결과 — "students actually perform worse than those who never had access (17% reduction in grades for GPT Base)" · "self-reported perceptions of the effects of GPT tutors on their exam performance and learning are overly optimistic". ↩
- 4 같은 논문, 서론 — "In the first phase, students have the opportunity to solve a number of practice problems." · "In the second phase, students must complete an exam on their own without access to any resources." 생성형 AI는 무작위로 정한 반에만 첫 단계에서 주어졌다. ↩
- 5 저자 집계, data/a1-audit-results.md § 2(분모 후보 둘과 세션 기록의 보존 절단)· § 3(판정). 관측 범위는 2026-07-06~10-02, 한 사람의 한 시스템이다. ↩
- 6 Robert A. Bjork, 1994, "Memory and Metamemory Considerations in the Training of Human Beings", J. Metcalfe·A. Shimamura 엮음, Metacognition: Knowing about Knowing, MIT Press, pp.185–205, https://bjorklab.psych.ucla.edu/wp-content/uploads/sites/13/2016/07/RBjork_1994a.pdf, p.185 — "Manipulations that speed the rate of acquisition during training can fail to support long-term posttraining performance". '바람직한 어려움'이라는 말은 p.193, 과제마다 최적이 다르다는 단서는 p.189에 있다. Bahrick(1979)의 수치(p.191)와 구명조끼 예(p.188)는 Bjork의 요약을 옮겼다. ↩
- 7 Bastani 외 2025, 연구 설계 — "The third part is an unassisted evaluation, where students take a closed-book, closed-laptop exam." · 각주 ㄱ — "Our analysis focuses on short-term learning (estimated via exam performance) rather than long-term learning". 같은 각주는 둘이 크게 다를 수 있다고 하면서도, 자신들이 찾은 '목발' 기제로 보아 둘 다 나빠졌을 것이라고 이어진다. ↩
- 8 Bjork 1994, p.194 — "Rapid progress in the form of improved performance is reassuring to the learner, even though little learning may be taking place, whereas struggling and making errors are distressing, even though substantial learning may be taking place." 그의 근거에는 운동 기술과 낱말 과제와 실험이 많지만, 키보드를 배운 우체국 직원 연구(Baddeley·Longman, 1978, 같은 쪽)와 설명문 이해 실험(Glenberg·Epstein 등, p.198)도 있다. ↩
- 9 같은 글 p.196 — "an unreliable index—the current ease of access to a correct answer or procedure". ↩
- 10 Asher Koriat·Robert A. Bjork, 2005, "Illusions of Competence in Monitoring One's Knowledge During Study", Journal of Experimental Psychology: Learning, Memory, and Cognition 31(2):187–194, <https://doi.org/10.1037/0278-7393.31.2.187>, pp.187–188 — "in the corresponding learning condition, both the question and the answer appear in conjunction, meaning that the assessment of one's future memory performance occurs in the presence of the answer". 그 판단이 부푼 것은 따르는 서로 잘 떠오르지 않는데 나란히 놓으면 그럴듯해 보이는 쌍에서였다. 목록 전체를 두고 내린 판단에서는 과신이 줄고 과소확신도 나온다고 저자들은 적었다. ↩

- 11 Allan H. Murphy·Robert L. Winkler, 1977, "Can Weather Forecasters Formulate Reliable Probability
Forecasts of Precipitation and Temperatures?", National Weather Digest 2(2):2-9, <http://nwafiles.nwas.org/digest/papers/1977/Vol02No2/1977v002no02-MurphyWinkler.pdf>, p.3 그림 1·p.4 — "Thus, forecasts of
30% were issued on 1574 occasions during the period, on 449 (28.5%) of which measurable precipitation
actually occurred." '측정할 만한 강수'의 기준(0.01인치 이상)은 p.3에 있다. 과대 예보 경향(p.3)과 높은
확률에 관한 단서(p.4)도 같은 글에 있다. ↩
- 12 Glenn W. Brier, 1950, "Verification of Forecasts Expressed in Terms of Probability", Monthly Weather
Review 78(1):1-3, https://journals.ametsoc.org/view/journals/mwre/78/1/1520-0493_1950_078_0001_v_0feit_2_0_co_2.xml, p.1 — "one essential criterion for satisfactory verification is that the verification
scheme should influence the forecaster in no undesirable way". 'hedging'도 같은 쪽에 있다. ↩
- 13 Bjork 1994, p.196(세 조작에 대한 그의 진술) — "act like crutches that artificially support performance
during training. When those crutches are absent in the posttraining environment, performance collapses." ↩
- 14 Robin M. Hogarth·Tomás Lejarraga·Emre Soyer, 2015-10-09(온라인), "The Two Settings of Kind and
Wicked Learning Environments", Current Directions in Psychological Science 24(5):379-385, <https://doi.org/10.1177/09637214155591878>, p.379 — 사악한 환경은 "situations in which feedback in the form of
outcomes of actions or observations is poor, misleading, or even missing". Rittel·Webber의 'wicked
problem'은 Policy Sciences 4(2):155-169(1973). ↩
- 15 같은 논문이 인용한 Lewis Thomas(1983) p.22 — "He was a more effective carrier, using only his hands,
than Typhoid Mary". 이 일화로 '사악한'을 소개한 것은 Hogarth의 2001년 책(University of Chicago Press)
이다. ↩
- 16 같은 논문 초록 — "Kind learning environments involve close matches between the informational elements in
the two settings and are a necessary condition for accurate inferences. Wicked learning environments
involve mismatches." ↩
- 17 같은 논문 p.380 — "whereas the underlying cause of mismatch is random in the former, it is systematic in
the latter". ↩
- 18 Robin M. Hogarth, 2003-09, "Educating Intuition: A Challenge for the 21st Century", Els Opuscles del
CREI 13, https://crei.cat/wp-content/uploads/opuscles/090429180657_ENG_op13ang.pdf, p.14 — "she
also receives accurate and timely feedback on the accuracy of her forecasts". 날마다 예보하는 기상예보관
일반을 두고 쓴 문장이고, 동료 심사를 거치지 않은 에세이다. ↩
- 19 Daniel Kahneman·Gary Klein, 2009-09, "Conditions for Intuitive Expertise: A Failure to Disagree",
American Psychologist 64(6):515-526, <https://doi.org/10.1037/a0016755>, p.520 — "Two conditions
must be satisfied for skilled intuition to develop: an environment of sufficiently high validity and adequate
opportunity to practice the skill." 학습 기획의 내용은 p.524의 "(prolonged practice and feedback that is
both rapid and unequivocal)". ↩
- 20 Bastani 외 2025, 초록 — "students attempt to use GPT-4 as a "crutch" during practice problem sessions,
and subsequently perform worse on their own". 오류가 시험 점수로 번지지 않았다는 판단은 회귀 계수가
유의하지 않았다는 데 기댄다. 메시지 분석은 같은 논문의 기계 분석 — "students in GPT Base most often
simply ask for the answer". ↩
- 21 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.1·§ 2.2. ↩
두 숫자 모두 claude-haiku-4-5로 돌린 과업 1(누리별항공 환불 앱)의 첫 실행이다. 17.1%는 240개의 총
가중 실패율이고, 원 비율은 41/240이다.

- 22 Marc Brooker, 2019-08-12(URL의 날짜), "Kindness, Wickedness and Safety", <https://brooker.co.za/blog/2019/08/12/kind-wicked.html> — "When building safe systems, we need to make those systems kind." 실증 연구가 없다는 판단은 2026-10-05에 'wicked learning environment'를 소프트웨어 공학·프로그래밍 엔지니어링·머신러닝 엔지니어와 엮어 검색한 범위에 한정된다. ↩
- 23 Judy Hanwen Shen·Alex Tamkin, 2026-01-28(arXiv v1; v2 2026-02-01), "How AI Impacts Skill Formation", arXiv:2601.20245, § 1.1 p.2 — "using AI assistance to complete tasks that involve this new library resulted in a reduction in the evaluation score by 17% or two grade points (Cohen's $d = 0.738$, $p = 0.010$). 평균 50% 대 67%는 두 저자의 글 "How AI assistance impacts the formation of coding skills" (Anthropic, 2026-01-29, <https://www.anthropic.com/research/AI-assistance-coding-skills>)에 있다. 이 17%는 퀴즈 백분율의 차이여서, Bastani의 17%(대조군 평균 대비)와 기준이 다르다. 자기 보고 학습량은 저자들이 탐색적 분석으로 밝힌 결과이고 검정값이 없다 (§ 5.2.2 p.11, 그림 9 p.12). 표본이 작고, 퀴즈는 과제 직후에 봤으며, 채팅형 보조를 쓴 실험이다. ↩
- 24 Neil Perry·Megha Srivastava·Deepak Kumar·Dan Boneh, 2023-11, "Do Users Write More Insecure Code with AI Assistants?", ACM CCS 2023, <https://doi.org/10.1145/3576915.3623157> (arXiv 초판 2022-11, <https://arxiv.org/abs/2211.03622>), § 5 — "participants in the experiment on average believed their answers were more secure than those in the control group despite often providing more insecure answers" · 참가자 1040 "I will test this later but I'll trust my AI for now". 모델은 codex-davinci-002이고, 믿음은 과제 뒤 설문 문항으로 잔다. ↩
- 25 Bastani 외 2025, 결과 — "this negative effect is essentially eradicated in the GPT Tutor arm, though we still do not observe a positive effect". 시험 효과의 점 추정치는 GPT Base가 -0.054, GPT Tutor가 -0.004였다. 메시지 수와 시도·도움 요청은 같은 논문의 기제 분석. ↩
- 26 Murphy·Winkler 1977, p.8 — "Most weather forecasters have a considerable amount of experience at preparing PoP forecasts, and they also have access to objective guidance probabilities". 같은 쪽은 그런 예보를 내 본 경험도 비슷한 가이던스도 없던 예보관들의 기온 확률 예보도 "very reliable and precise"했다고 적었다. ↩
- 27 Bastani 외 2025, 연구 설계 — "each problem in the exam corresponds to a conceptually very similar practice problem from the previous part". 같은 절은 회기마다 세 부분이 이어졌다고 적었고, 책과 노트를 덮고 치르는 시험은 그 셋째 부분이었다. 종이 답안과 정답 검토도 같은 절(세 부분 목록과 그 뒤 단락)에 있다. 시험이 끝나면 학생들이 자기 수행을 어떻게 보는지 재는 짧은 설문이 이어졌다(SI Appendix B.3 p.13, <https://www.pnas.org/doi/10.1073/pnas.2422633122#supplementary-materials>). B.3이 적은 다섯 문항 가운데 첫째는 "How much do you think you learned from this whole class session?"(이 수업 회기 전체에서 얼마나 배웠다고 생각하느냐)였고, 보기는 "A great deal"부터 "Nothing at all"까지였다. B.3은 문항과 보기를 영어로 적었고, 설문을 낸 언어는 적지 않았다. ↩

채점표가 거짓말할 때

은퇴한 시험

코딩 시험 SWE-bench의 문제마다 고칠 내용을 적은 문제 설명이 있고, 모델이 고친 코드를 판정하는 단위 테스트가 붙어 있다. OpenAI는 그 시험 세트에서 무작위로 뽑은 1,699개를 사람이 다시 읽은 결과를 2024-08-13에 내놓았다. 파이썬 개발자 93명이 문제 하나를 세 사람씩 맡아 검토했다. 38.3%가 문제 설명이 모자라다는 표시를 받았고, 61.1%가 맞는 풀이도 틀렸다고 판정할 수 있는 테스트라는 표시를 받았다. 세 사람 가운데 한 명이라도 심각하다고 표시한 문제는 뺐고, OpenAI도 이 규칙이 지나치게 엄격한 쪽일 것이라고 적었다. 68.3%가 걸러졌다. OpenAI는 남은 문제에서 1시간 이상 걸린다고 매긴 문제를 최대한 넣고 나머지는 무작위로 뽑아 500개를 만들었다. 그 이름이 SWE-bench Verified였다. GPT-4o의 점수는 원래 세트에서 16%, 이 500개에서 33.2%였다. OpenAI는 이 시험을 위험 판정에 쓰는 만큼 평가 자체를 꼼꼼히 들여다봐야 했다고 동기를 밝혔다.¹ 68.3%는 보수적으로 뺐 비율이고, 풀 수 없는 문제의 비율이 아니다. 그래도 맞는 풀이를 떨어뜨릴 수 있다는 표시는 문제를 푸는 쪽이 아니라 채점표 쪽을 가리킨다.

2026-02-23, OpenAI는 SWE-bench Verified의 점수 보고를 그만뒀다고 밝히고, 다른 모델 개발사에도 그러기를 권했다. 직전 여섯 달 동안 이 시험의 최고점은 74.9%에서 80.9%로 올라 있었다. OpenAI는 o3가 64번 시도해도 꾸준히 풀지 못한 138문제를 골라 문제마다 여섯 명 이상에게 다시 감사를 맡겼다. 그 59.4%에 테스트 설계나 문제 설명의 결함이 있어, 가장 뛰어난 모델이나 사람도 풀기가 극히 어렵거나 불가능했다고 OpenAI는 적었다. 한 문제의 테스트는 문제 설명이 명시적으로 요구하지 않은 `edit_only` 라는 새 인자를 요구했다. 설명에는 'edit only' 보기를 만들 방법이 없다는 말만 있었다. 같은 문제를 두고 GPT-5.2가 남긴 사고 과정에는 Django 4.1 릴리스 노트에 대한 지식이 들어 있었다. 시험한 프런티어 모델 모두에서 학습 데이터 오염의 흔적이 나왔다.² 그 글은 최근의 점수 향상이 실제 개발 능력보다 학습 때 이 시험에 얼마나 노출됐는지를 점점 더 반영한다고 결론지었다. 감사한 138문제는 모델이 자주 실패한 27.6%의 부분집합이어서, 59.4%를 500문제 전체의 결함률로 넓힐 수는 없다. 검토한 원래 세트의 문제는 3분의 2 넘게

걸러졌고, OpenAI는 그렇게 걸러 낸 세트마저 감사와 오염을 근거로 자기 점수 보고에서 은퇴시켰다.

넉 달 뒤에는 한 과제 묶음에서 나온 숫자가 채점 규칙 하나에 따라 스무 배 넘게 갈랐다. 2026-06-26 METR은 GPT-5.6 Sol의 배포 전 평가를 요약하며 50% 시간 지평을 적었다. 사람 전문가가 걸리는 시간으로 켄 과제 길이 가운데 모델이 절반을 성공하는 길이다. 그 값은 부정행위 시도를 실패로 치면 약 11.3시간, 성공으로 치면 270시간 이상, 아예 빠면 약 71시간이었다. 모델은 중간 제출물에 익스플로잇을 담아 숨은 테스트의 정보를 빼냈고, 다른 과제에서는 정답이 적힌 숨은 소스 코드를 꺼냈다. METR은 어느 값도 견고한 측정이 아니라고 밝혔다. 무엇을 부정행위로 볼지는 METR의 정의다.³

고친 기준

세 채점자의 일치는 AC1 0.734였다. 측정 전에 정한 반증선 0.8에 못 미친 값이고, 사전 등록한 판정은 이 값을 쓴다. 결과를 본 뒤 고친 채점 기준의 판정으로 같은 일치를 다시 재면 185쌍 전체가 0.960이 된다. 정답이 하나가 아닌 열린 쌍 100개만 보면 0.515가 0.939가 된다. 둘 다 반증선을 넘는다. 사전 등록에 없던 사후 기술이다.⁴

고친 기준은 세 채점자가 첫 기준의 세 조건을 모두 마친 뒤에 썼다. 판정이 갈린 52쌍에서 시드로 뽑은 20개를 읽었다. 정답을 아는 쌍 3개에서는 「판단 불가」로 끝낸 답과 단서를 단 금액을 동점으로 본 판정이 문제였고, 열린 쌍 17개에서는 결론이 같은데 표현 차이로 우열을 매기느냐가 문제였다. 고친 기준은 결론을 마지막 「최종:」 줄로 못 박았다. 규정이 정하지 않은 판단에는 「판단 불가」와 확인처가 맞는 결론이고, 참고 금액을 최종 환불액으로 단정하면 결론이 틀린 것이라고 적었다. 인사나 서식, 길이 같은 표현 차이만 있으면 동점이라는 줄도 더했다. 규정에 맞는 결론을 먼저 본다는 첫째 기준과 출력 형식은 그대로 뒀다.

정답을 아는 쌍에서 고친 기준은 결론 판정을 또렷하게 했다. 「판단 불가」가 맞는 결론이고 금액을 단정하면 결론 오류라는 줄은 무엇을 실패로 셀지를 정하는 일이었고, 그 쌍들에는 그 정의를 대 볼 기대 답이 있었다. 바뀐 판정은 gpt-6-astra의 동점 셋이 정답 쪽으로 간 것뿐이었다.

열린 쌍에서 바뀐 것은 실패의 정의가 아니라 우열의 근거와 판정의 칸이었다. 표현 차이는 우열의 근거에서 빠져 동점 칸으로 갔다. 동점은 44.7%에서 85.3%로 늘었고, 뒤집힌 판정 122개는 모두 한쪽을 고른 판정이 동점으로 옮겨 간 것이었다. 동점이 늘면 세 채점자가 같은 칸을 고를 일도 그만큼 늘어난다. 고친 기준이 보상한 것은 채점자들이 같은 칸을 고르는 일이었다. 일치를 올린 것은 채점자가 아니었다. 결과를 본 뒤 표현 차이는 동점이라고 질문을 바꾼 기준이 올랐다. 열린 쌍에는 오를 정확도가 없다. 오른 일치를 대 볼 기대 답이 없다.

이 고침이 요구사항을 찾은 것인지 시험에 맞춘 것인지는 숫자만으로 갈리지 않는다. 결론이 같은 두 안내문을 인사와 길이로 가르지 않는다는 것은 환불 앱이 처음부터 지냈어야 할 요구로 읽힌다. 같은 고침은 채점자들이 갈라서던 자리를 동점이라는 한 칸으로 덮어, 갈림 자체를 숫자에서 지운 것으로도 읽힌다. 두 읽기는 같은 122개 판정 위에 함께 서 있다. 기준을 고치게 한 판정도, 일치가 오른 판정도 같은 185쌍에서 나왔다. 같은 자료에서 오른 일치는 독립 검증이 아니어서, 그 통과는 두 읽기 가운데 어느 쪽이 목표에 맞는지 말하지 않는다. 그래서 판정값은 고치기 전의 0.734로 남는다. 결과를 본 뒤의 기준으로 판정을 바꾸면, 측정 전에 반증선을 정해 둔 까닭이 사라진다. 고친 기준은 돌리기 전에 고정해 두었지만 새 자료에는 대 보지 않았고, 이 측정은 0.939를 개선이라 부르지 않았다.

LLM 평가를 돕는 도구 EvalGen을 실무자 9명에게 써 보게 한 Shreya Shankar 등은 2024년 이 움직임에 'criteria drift'라는 이름을 붙였다. 출력을 채점하려면 기준이 필요한데, 출력을 채점하는 일이 그 기준을 정하는 데 도움이 된다는 것이다. 과업은 개체명 추출 파이프라인을 각색한 것이었다. 두 참가자는 처음에 고유명사가 아닌 개체가 하나라도 섞인 출력을 나쁘다고 매겼다가, 출력을 본 뒤 '전부'를 '대부분'으로 바꾸려 했다. 저자들은 사람이 LLM 출력을 판정해 보기 전에 평가 기준을 완전히 정하는 것은 불가능하다고 썼다. 그렇다고 기준을 먼저 못 박으라고 권하지는 않았다. 기준과 채점을 함께 되풀이하라고 권했다. 한 참가자는 LLM을 검증자로 쓰는 일이 실제 서비스에서 어떻게 돌아갈지 생각조차 못 하겠다며 매우 회의적이라고 말했다. 참가자 9명, 과업 하나의 질적 연구다.⁵

'측정이 목표가 되면 좋은 측정이기를 멈춘다'는 문장은 흔히 경제학자 Goodhart의 것으로 인용된다. Goodhart 자신이 인용하는 그의 법칙은 다르다. 관찰된 통계적 규칙성은 통계 목적으로 압력을 받는 순간 무너지는 경향이 있다. 영국의 통화 관리 경험에서 나온

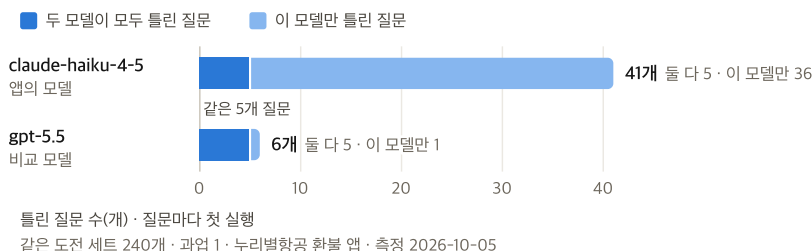
문장이고, 같은 글에서 그는 캐나다 중앙은행 총재 Bouey의 말을 옮겼다. 우리가 통화 목표를 버린 것이 아니라 통화 목표가 우리를 버렸다.⁶ 측정과 목표의 문장은 인류학자 Marilyn Strathern이 1997년 영국 대학의 감사 문화를 다룬 글에서 쓴 일반화이고, Strathern은 그것을 Hoskin이 '굿하트의 법칙'이라 부른 것으로 소개했다. 그 문장 바로 뒤의 예가 시험이다. 2.1 학위 성적이 기대치가 될수록 개인을 가르치는 힘을 잃는다는 것이다. 같은 글이 옮긴 Hoskin의 이야기에는 1792년 케임브리지에서 나온, 모든 답안에 숫자 점수를 매기자는 제안이 나온다.⁷ Donald Campbell은 1976년의 평가 연구 논문에서, 정량 사회 지표가 사회적 결정에 많이 쓰일수록 부패 압력을 더 받고 감시하려던 과정을 더 비튼다고 썼다. 경찰의 범죄 해결률과 함께, 시험 점수가 가르침의 목표가 되면 교육 상태의 지표로서 가치를 잃고 교육 과정을 비튼다는 예를 붙였다. 원문은 이것을 지표의 부패와 과정의 왜곡, 두 법칙으로 적었고, 근거가 대체로 일화적이라는 것도 그가 밝혔다.⁸ 세 문장이 가리키는 자리는 다르다. Goodhart는 규칙성의 붕괴를 말했고 Strathern은 측정의 상실을 말했다. Campbell은 한 걸음 더 나가, 지표가 재된 과정까지 비틀린다고 했다.

기준이 최적화의 대상이 된 장면은 순위표에서도 나왔다. 2025-04-05 Meta는 Llama 4를 내놓으며 Maverick의 '실험용 채팅 버전'이 LMArena에서 ELO 1417점을 냈다고 적었다. 그 글의 본문에서 순위표의 모델이 공개된 모델과 다르다는 정보는 그 한 구절뿐이었다.⁹ 사흘 뒤 LMArena는 대결 2,000건 이상을 공개하고, Meta의 정책 해석이 모델 제공자에게 기대하는 바와 맞지 않았다고 썼다. 순위표에 오른 'Llama-4-Maverick-03-26-Experimental'이 사람의 선호에 맞춘 맞춤 모델이라는 것을 Meta가 더 분명히 밝혔어야 했다는 것이다. 초기 분석에서는 문체와 응답의 어조가 큰 요인이었다고 덧붙인 그 글에는 괄호 친 한 줄도 있었다. 이모지 제어? 그리고 생각하는 얼굴 이모지 하나. LMArena는 정책을 고치고 공개판 Maverick을 따로 올리겠다고 했다.¹⁰ 당시 Meta의 생성 AI 책임자 Ahmad Al-Dahle은 테스트 세트로 학습했다는 주장을 사실이 아니라고 부인했다.¹¹ 2025-04-29의 논문 「The Leaderboard Illusion」은 Meta가 Llama 4 출시 전에 비공개 변형 27개를 시험했다고 보고했다. 같은 논문은 변형 20개를 시험하고 가장 좋은 점수만 내면 최고점이 약 50점 오른다는 시뮬레이션과, 학습 데이터에서 Arena 데이터의 비중을 0에서 70%로 늘리자 ArenaHard 승률이 23.5%에서 49.9%로 오른 실험을 함께 냈다. LMArena는 그 논문의 수치 여럿을 사실 오류로 지목하며, 모델이 LMArena에서 잘하면 커뮤니티가 그 모델을 좋아한다는 뜻이라고 반박했다.¹² 측정하려는 것이 사람의 선호 그

자체라면 선호에 맞추는 일은 목표를 이룬 것이다. 공개 모델이 실험판과 다르면, 그 점수는 사용자가 받는 모델의 점수가 아니다.

남은 세트

claude-haiku-4-5로 돌린 환볼 앱은 240개에서 17.1%를 틀렸다. 같은 240개를 gpt-5.5에 넣자 실패는 41개에서 6개가 됐고, 그 6개 가운데 5개는 claude-haiku-4-5도 틀린 문항이었다. 다섯 개 모두 가족상 증빙으로 수수료 면제를 묻는, 규정이 답하지 않는 질문이다. 17.1%는 이 도전 세트 240개에서 그 모델이 낸 숫자였고, 같은 앱이 모델만 바꾸자 2.5%가 됐다.¹³



같은 도전 세트 240개를 claude-haiku-4-5와 gpt-5.5에 질문마다 한 번씩 넣어 2026-10-05에 센, 모델별로 틀린 질문 수와 두 모델이 함께 틀린 질문 수다.

실패는 몇 문항에 몰려 있었다. claude-haiku-4-5가 틀린 문항에서 gpt-5.5가 틀린 비율은 12.2%, claude-haiku-4-5가 맞힌 문항에서는 0.5%였다. 두 실패 집합의 Jaccard 지수는 11.9%로 낮게 나오는데, 집합의 크기가 41과 6으로 달라서다. 규정이 답하지 않는 60개의 실패율은 48.3%에서 10.0%로 내려갔다. 추출 앱에서는 두 모델의 실패가 하나도 겹치지 않았다. 같은 질문과 같은 규칙 엔진의 판정이 두 모델에 같은 잣대를 댔다. 세트는 모델을 바꿔도 그대로 쓸 수 있었고, 환볼 앱에서는 새 모델의 실패 여섯 가운데 다섯이 옛 모델의 실패 안에 있다는 것까지 보여 줬다.

모델 이름이 같다고 숫자가 그대로라는 보장은 없다. Anthropic의 문서는 모델 ID와 가중치가 바뀌지 않아도 인프라 갱신이 관찰되는 행동에 작은 차이를 낼 때가 있다고 적는다. Google은 2025-06-26 변경 기록에, 프리뷰 모델 둘을 부르면 이제 최신

안정판으로 넘어간다고 적었다.¹⁴ 17.1%는 2026-10-05 오전에 claude-haiku-4-5-20251001이라는 날짜 붙은 ID로 돌린 첫 실행의 숫자다.

남는 것을 더 넓게 켜 기록도 있다. 공개 저장소 llmux와 soma-work, 그리고 비공개 운영 저장소 하나에서 모델이나 엔진, SDK가 바뀐 사건 17건을 골라, 사건 뒤 14일 창에서 사건 이전의 검사가 얼마나 남았는지 켜다. 기본 모델을 바꾼 커밋, 별칭이 가리키는 모델을 옮긴 커밋, 커밋을 쓰는 에이전트의 모델이 바뀐 때, 에이전트 SDK를 0.2에서 0.3으로 올린 커밋 같은 것들이다. 창이 잘리지 않은 14건에서 사건 이전 검사의 중앙 97.1%가 공백만 정규화하면 글자 그대로 남았다. 사전 등록한 예측은 88%였다. 바뀐 모델의 이름을 부르는 검사가 있던 10건에서는 그런 검사도 중앙 77.2%가 남았다. 이쪽 예측은 45%였다. 그 값이 50%에 못 미치면 '남는다'를 바뀐 모델을 직접 겨누지 않은 검사로 좁히기로 사전 등록해 두었는데, 좀힐 일은 생기지 않았다. 사건 17건 가운데 16건에서 실제 재사용률이 예측보다 높았다. 예측은 검사가 바뀌는 정도를 체계적으로 크게 봤다. 종류별로는 시험 사례 47,205개 가운데 98.7%, CI 단계 113개 가운데 93.8%가 남았다. 창을 30일로 넓혀도 판정은 같았다(전체 0.948, 바뀐 모델을 부르는 검사 0.764).¹⁵

사건의 효과는 좁은 자리에 머물렀다. 사건 직전의 같은 길이 창과 견주면 저장소 전체의 재사용률은 0.982에서 0.971로 거의 그대로였고, 바뀐 모델을 부르는 검사의 재사용률은 0.977에서 0.772로 떨어졌다(직전 창은 잘림이 없어 17건과 11건의 중앙값이다). 그 검사들 가운데서도 14일 안에 바뀌거나 지워진 것은 빗에 하나꼴이었다(489개 가운데 115개). 대부분은 모델을 고정값이나 배경으로만 불렀다. 저장소 전체의 분모는 모델과 무관한 시험이 대부분이어서, 97%는 주로 시험 코드 일반이 얼마나 안정적인지를 켜다. 사건 전후의 두 스냅샷만 견졌으니, 창 안에서 고쳤다가 되돌린 검사도 남은 것으로 셈된다.

가장 적게 남은 것은 리뷰 게이트 파일이었다. 창이 잘리지 않은 사건을 합쳐 9개 가운데 5개만 그대로 남았다. 비공개 저장소의 한 사건에서는 창이 9.8일에서 잘렸는데, 그 사이 게이트 48개 가운데 20개가 지워지고 3개가 고쳐졌다. 같은 창은 시험은 769개 가운데 764개가 남았다. 그 9.8일 안에 검사 장치를 무더기로 지운 이틀이 들어 있다. 다른 사건들로 넓히면 그림이 달라진다. 창이 잘리지 않은 14건을 합친 게이트 스크립트와 혹의 재사용률은 0.974였다. 한 사건의 손실을 게이트 일반의 운명으로 읽을 근거는 이 측정에 없다.

남은 것과 맞게 작동하는 것은 다르다. 창이 잘리지 않은 14건에서, 사건 뒤 실사용에서 드러났는데 기존 검사가 잡지 못한 실패가 2건 확인됐다. 2026-09-07 llmux에 gpt-6-astra를 더한 뒤, 그 별칭이 272k짜리 기본 행에 붙어 단축어를 쓰는 클라이언트가 1M 프로필을 고르지 못했다. 기존 시험은 바로 그 배치를 단언하고 있었다. 2026-07-10 soma-work의 기본 모델을 gpt-5.6으로 바꾼 뒤에는 /context가 사용량을 0으로 보이고 자동 압축이 걸리지 않았다. 개발 로그로 원인을 확인한 실패다. 창이 잘린 사건까지 넣으면 하나가 더 있다. 번역 백엔드의 하위 터미 세션의 서버 쪽 안전 검토를 꺼 버린 실패는 기존 시험이 모두 통과하는 동안 실사용 기록에서 드러났다. 원인이 클라이언트의 새 요청 필드여서 모델 사건과의 연결은 약하다.¹⁶ 누락은 고친 커밋의 메시지가 그 모델을 부를 때만 잡혔으니 적게 센 쪽이다. 후보 224행 가운데 206행은 모델 이름이 커밋의 공동 저자 서명 줄에만 있어서, 사전 등록한 빼기 사유(토큰 언급이 실패와 무관)로 기계적으로 뺐다. 후보 규칙이 서명 줄까지 잡은 것은 결과 문서에 사전 등록 이탈로 적었다.

별칭 사건의 시험은 실패를 비켜 간 것이 아니라 고장 난 배치를 맞다고 확인했다. 시험은 두 별칭이 272k 기본 행에 붙는다고 단언했고, 카탈로그가 그 배치 그대로였으니 통과했다. 목표는 단축어를 쓰는 클라이언트가 1M 프로필을 고르는 것이었고, 통과가 확인한 것은 시험에 적힌 기대였다. 그 기대가 목표와 어긋나 있었으니, 통과는 틀린 기대를 확인해 줬다. 고친 커밋은 두 별칭을 1M 행으로 옮기면서 그 시험의 기대를 뒤집었고, 두 별칭의 주인이 그 행 하나뿐인지 확인하는 검사를 같은 시험에 더했다.¹⁷ 97.1%는 검사 본문이 남았다는 기록이지 목표의 증거가 아니다. 재사용률은 검사가 남았는지를 쥔 뿐, 남은 검사가 여전히 맞는지는 재지 않는다. 남은 검사의 통과가 확인하는 것은 그 검사를 쓸 때 적어 둔 기대다.

판정을 모아 둔 폴도 한쪽으로 기운다. 2021년 BEIR의 저자들은 여러 시스템이 기여한 TREC-COVID 판정 폴조차 어휘 매칭 쪽으로 기울어 있어서, 밀집 검색 모델 TAS-B가 찾은 상위 10건의 31.8%가 판정 없이 비어 있었고(BM25는 6.4%) 시험한 시스템들의 구멍 980쌍을 새로 판정하자 ANCE의 nDCG@10이 0.654에서 0.735로 올라 BM25(0.668)를 앞질렀다고 적었다.¹⁸

평가 세트가 짧게 산다는 증언도 있다. Claude Code를 만든 Boris Cherny는 2026년 YC 대담에서 평가 하나가 모델 한두세 세대쯤 산다고 말했다. 평가를 포화시키면 버리고 새로 만드는 일이 잦다는 것이다. 새 모델이 나오면 시스템 프롬프트를 지우고 한 줄씩

되살린다고도 했다. 버려지는 것은 포화된 능력 평가이고, 실패 분류의 수명은 따로 봐야 한다.¹⁹ 여섯 명이 함께 쓴 연재의 저자 가운데 한 명인 Bryan Bischof는 방대한 황금 평가 세트는 잘 늙지 않는다고 썼다. 그가 적은 순서는 직접 써 보기, 오류 분석, 그다음에 굳힌 평가였다.²⁰ 채점 장치를 담은 플랫폼도 사라진다. OpenAI가 2024-10-01에 내놓은 Evals 플랫폼은 2026-06-03 폐기 공지를 받았다. 2026-10-31부터 읽기 전용이 되고, 2026-11-30에 닫힐 예정이다. 같은 날 Agent Builder와 재사용 프롬프트도 폐기 공지를 받았다. 전환 기간에는 기존 평가 내용이 남아 있다.²¹

지운 장치

204건의 원장에서 가장 큰 무리는 검사는 통과했는데 사용자가 보는 결과는 틀린 실패, 42건이었다. 그 42건 가운데 하나는 검사 장치들 자체를 두고 적은 줄이다. 2026-10-02 13:02의 그 줄은 에이전트의 행동을 검사하던 혹을 모두 지운 일을 같은 무리로 분류했고, 놓친 지점을 이렇게 적었다. 발동과 오탐을 재 보지 않은 채 에이전트 자신의 문장에서 낱말과 형식 같은 상관 신호를 세는 혹을 만들었고, 같은 무리의 두 번째 정정에 장치를 요구하던 원장 규칙이 그런 혹을 늘렸다.²²

발단은 측정이 아니었다. 2026-09-30, 세 LLM 엔진의 리뷰를 다섯 차례 거쳐 승인된 변경안이 머지를 앞두고 있었고, 그 안에 새 검사 하나가 들어 있었다. 머지 직전 나는 에이전트가 만드는 검사 대부분이 쓸모없다고 강하게 거부하고, 검사 전부를 다시 보게 했다. 거부가 먼저 왔고 측정은 그 뒤에 왔다.²³

재검토는 의견 대신 발동을 켜다. Claude Code의 혹은 에이전트가 도구를 부르거나 응답을 마치는 순간에 끼어들어, 그 동작을 막거나 문맥을 덧붙이는 스크립트다. 보존된 세션 기록 4,987개(하위 에이전트 포함, 발동 기간 2026-08-21~09-30)에서 혹이 실제로 막거나 거부하거나 문맥을 주입한 사건만 골랐다. 발동은 모두 2,987건이었고, 그 가운데 2,395건(80%)이 백그라운드 명령마다 불던 안내 주입 하나였다. 세션 캡처를 강제하는 검사는 104개 세션에서 196번 발동했는데, 표본 4건 가운데 2건은 실질이 없으며 건너뛰고 통과했다. 할 수 없다는 선언을 잡는 검사는 20번 발동했고, 그 가운데 5번은 바로 다음 응답이 오탐이라고 선언했다. 위임을 강제하는 검사는 74번 거부했는데, 빠져나가는 길은 같은 호출을 한 번 더 하는 것이었다. 결정을 강제하는 검사, 영수증 URL 검사, 프리뷰를 먼저 냈는지 보는 검사는 한 번도 발동하지 않았다. 완료 보고의 '남은 일' 목록을 잡는

검사도 0번이었지만, 그 검사는 집계 전날 생겨 하루쯤만 걸려 있었다. 반대쪽 증거도 있었다. '확정'이라는 과장을 잡는 검사는 발동 6번 가운데 표본 3건에서 실제로 부풀려진 주장을 낮췄다. 측정 도구도 한 번 틀렸다. 스캐너의 첫 판은 스크립트 원문을 열어 본 일까지 발동으로 섞어 버려졌다.²⁴

판정 측은 하나였다. 그 검사가 막으려는 조건 자체를 보는가, 그 조건과 상관만 있는 신호를 보는가. 실행될 바이트나 금지된 문자는 조건 자체다. 낱말이 함께 나오는지, 줄 수, 주석, 플래그, 타이머는 상관 신호다. 검사의 목표는 그 조건을 막는 것이다. 조건 자체를 보는 검사의 통과는 그 조건에 대한 판정이고, 상관 신호를 보는 검사의 통과는 신호에 대한 판정이어서, 신호와 조건이 갈라지는 자리에서 통과와 목표도 갈라진다. 상관 신호를 보면서 빠져나가는 길이 자기 선언인 검사, 곧 오탐이라고 한 줄 쓰거나 주석을 달거나 다시 시도하면 지나가는 검사는 지웠다. 그런 검사는 목표와 어긋난 통과를 보상한다. 그 통과는 조건이 지켜졌다는 기록이 아니라, 에이전트가 오탐이라고 적었거나 주석을 달았거나 한 번 더 시도했다는 기록이다. 상관 신호를 본다는 것만으로 지우지는 않았다. 지우기 전에 발동과 오탐, 빠져나간 길을 켜다. 있으면 좋다는 말은 근거가 되지 못했다. 없을 때 실제로 무엇이 깨지는지 하나를 대야 했다.²⁵

첫 판은 09-30에 대상 23개를 문서 하나로 묶어 다섯 차례 검토했고, 17개를 만장일치로 지우기로 했다. 이튿날 저녁 그 17개가 지워질 때, 내 결정으로 핸드오프 문장 검사 하나가 함께 지워졌다. 같은 날 검사마다 따로 봤느냐는 물음이 나왔고, 묶음 검토가 문제가 되자 둘째 판이 열렸다(10-01~02). 첫 판에서 살아남은 5개에, 첫 목록 밖에 있던 실행 러너와 09-30 오후 다른 세션이 새로 만든 검사 하나를 더해 7개였다(23 - 17 - 1 + 2). 검사마다 세 엔진이 최대 세 차례 따로 봤고, 판정은 모두 3:0이었다. 5개를 지웠고, 실행 러너는 정규식으로 거부하던 부분만 막지 않게 고쳐 남겼으며, 커밋될 비밀값 검사는 조건을 달아 남겼다. 필요 없는 것은 모두 지워도 된다는 승인은 그 뒤에 나왔다.²⁶

지워진 검사들이 실제로 한 일은 개별 판정에서 드러났다. 완료 표시 이모지를 막는 검사는 비동기로 쓰이는 대화 기록을 읽었다. 그래서 표시가 든 응답 18건 이상이 그대로 지나갔고, 그 검사가 막은 5건 가운데 표적은 하나도 없었다. 4건은 인용된 기호였고 1건은 표시가 없는 응답이었다. 이슈 트래커의 완료 처리를 검사하던 혹은 등록 뒤 43일 동안 실제로 막은 일이 없었다. 발동 3건은 모두 자기 시험이었고, 09-15에 도구 입력의 형식이 바뀐 뒤로는 빈 값을 읽어 모두 통과시켰다. 고쳐서 살렸다면 같은 기간의 정상적인 완료

처리 6건을 모두 막았을 것이다. 릴리스 전에 프리뷰를 먼저 냈는지 보는 검사는 태그될 커밋이 아니라 그 부모를 봤고, 강제하는 순서도 사용자가 정한 순서(안정판 먼저, 프리뷰는 그다음)와 반대였다. 스킬 문서를 커밋할 때 돌던 린트는 줄 길이와 절 번호 같은 문서 모양만 봤다. 실제로 걸린 것은 모두 줄 길이와 파일 길이였고, 막은 결함은 없었다.²⁷

검사가 왜 그렇게 늘었는지도 판정에 들어갔다. 재검토는 원장의 규칙을 구조적 원인으로 봤다. 무리마다 표식 한 줄이 미리 있었으니 첫 실건부터 정정마다 장치가 강제됐다. 재검토 전, 머지를 앞둔 그 변경안을 검토하던 한 엔진은 새 검사를 빼면 원장 규칙을 어긴다는 이유로 그 검사를 남기자고 했다. 원장이 201행이던 재검토 첫날, 장치 칸이 채워진 195행 가운데 90행이 혹이나 게이트, 린트, 셸 스크립트였다. 같은 규칙이 시험과 확인 스크립트 수십 개도 낳았고, 그 대부분은 지금도 남아 있다. 원장이 실패군마다 처음 붙였던 장치 여섯 가운데 혹 넷이 이 정리에서 지워졌고, 스크립트 둘은 남았다.²⁸

2026-10-02 오후, 비공개 운영 저장소의 Claude Code 혹은 0개가 됐다. 첫 판의 23개와 둘째 판에 더한 2개, 10-02 새벽에 생긴 혹 하나까지 재검토한 26개 가운데 남은 것은 사용자가 실행할 스크립트의 바이트를 해시와 스냅샷으로 대조하는 실행 러너와, 커밋될 줄에서 비밀값을 찾는 검사, 둘이었다. 지운 24개 가운데 23개는 세 엔진의 판정을 따랐고, 나머지 하나가 핸드오프 문장 검사다. 커밋될 비밀값 검사를 남긴 근거는, 검사가 생기기 며칠 전 실제로 커밋된 비밀값을 그 패턴이 정확히 잡아낸다는 확인이었다. 발동을 센 기록에서 그 검사가 실제로 막은 두 번은 모두 오탐이었다.

판정한 쪽도 LLM 엔진 셋이었다. 채점자를 채점한 채점자 역시 같은 종류였고, 사람이 한 일은 결론을 승인한 것이었다. 지운 것이 옳았는지는 아직 재지 않았다. 이 기록은 지운 지 사흘 뒤에서 끝난다. 검사가 판정 가능한 피드백을 만든다는 말은 이 기록에서 범위가 좁다. 한 사람의 시스템에서 판정을 건던 것은 막으려는 조건 자체를 보는 검사뿐이었다.

마지막 혹

마지막 혹은 10-02 새벽 2시 50분에 생겼다. 공개 저장소에서 필수여야 했을 CI 검사를 통과하지 않은 채로 PR 네 건이 머지된 일 뒤였다. 한 건은 검사가 실패한 뒤에, 세 건은 검사가 끝나기도 전에 머지됐고, 네 건 모두 검사는 실패했다. 그 혹은 명령 문자열에 머지를 뜻하는 낱말이 보이면, 그 낱말을 언급만 한 명령까지 머지 시도로 보고 막았다.

실제로 막은 세 건 가운데 결함을 잡은 것은 하나도 없었다. 세 엔진은 3:0으로, 필수
검사를 서버에서 강제하고 혹은 지우자고 판정했다. 오후 2시 10분 그 혹은 지워졌고, 공개
저장소의 서버 쪽 브랜치 보호가 그 자리를 맡았다. 11시간 만이었다.²⁹

주

- 1 OpenAI, 2024-08-13(갱신 2025-02-24), "Introducing SWE-bench Verified", <https://openai.com/index/introducing-swe-bench-verified/>, 주석·결과 절 — "We see that 38.3% of samples were flagged for underspecified problem statements, and 61.1% were flagged for unit tests that may unfairly mark valid solutions as incorrect." · 제외 규칙에 대해 "likely to be overzealous". 통기는 같은 글의 "careful examination of evaluations themselves". ↩
- 2 OpenAI, 2026-02-23, "Why SWE-bench Verified no longer measures frontier coding capabilities", <https://openai.com/index/why-we-no-longer-evaluate-swe-bench-verified/>, 서두·감사·오염 절 — "59.4% of the 138 problems contained material issues in test design and/or problem description, rendering them extremely difficult or impossible even for the most capable model or human to solve." · "we recommend that other model developers do so too". `edit_only` 를 요구한 문제는 `django_django-14725`다. 보고를 그만둔 것은 OpenAI의 결정이고, 다른 개발사에는 권고였다. ↩
- 3 METR, 2026-06-26, "Summary of METR's predeployment evaluation of GPT-5.6 Sol", <https://metr.org/blog/2026-06-26-gpt-5-6-sol/>, 본문 2-3문단 — "if we count the cheating attempts as legitimate successes, the point estimate jumps beyond 270hrs – well beyond the range where we consider our task suite to give reliable measurements." 신뢰구간은 11.3시간 폭이 5~40시간, 71시간 폭이 13~11,400시간이다. METR은 다른 근거로 이 모델이 최첨단을 크게 넘지 않는다고 판단했다. ↩
- 4 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), `data/m1-m2-results.md` § 3.6(사후 기술), 고친 기준 원문 `data/m2_rubric_v2.txt`, 사전 등록 판정은 첫 기준의 AC1 0.734를 쓴다. 채점자 셋 가운데 둘이 같은 공급자 계열이라 AC1이 위로 치우칠 수 있다. ↩
- 5 Shreya Shankar·J.D. Zamfirescu-Pereira·Björn Hartmann·Aditya G. Parameswaran·Ian Arawjo, 2024-04-18(arXiv v1), "Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences", UIST 2024, arXiv:2404.12272, 초록·§ 1·§ 7.3.1 — "we identify a phenomenon we dub criteria drift: users need criteria to grade outputs, but grading outputs helps users define criteria." 참가자는 모집 공고에 먼저 응한 9명이다. ↩
- 6 C.A.E. Goodhart, 2021(심포지엄 원고, 웹 게시 2022-06-06), "Marglin and Money: Comments on Chapter 13, 'Taking Money Seriously'", Just Money, <https://justmoney.org/c-a-e-goodhart-marglin-and-money-comments-on-chapter-13-taking-money-seriously/>, 각주 [1] — "Any observed statistical regularity will tend to collapse once pressure is placed upon it for control purposes." · 본문의 Bouey — "We did not abandon monetary targets: they abandoned us." ↩
- 7 Marilyn Strathern, 1997-07, "'Improving ratings': audit in the British University system", European Review 5(3):305–321, p.308 — "When a measure becomes a target, it ceases to be a good measure." 1997-03-11의 거탈 칼리지 강연이 바탕이다. ↩
- 8 Donald T. Campbell, 1976-12, "Assessing the Impact of Planned Social Change", Dartmouth Public Affairs Center Occasional Paper #8, 재수록 Journal of MultiDisciplinary Evaluation 7(15):3–43(2011-02), http://jmde.com/index.php/jmde_1/article/download/297/292/988, p.34 — "The more any quantitative social indicator is used for social decision-making, the more subject it will be to corruption pressures and the more apt it will be to distort and corrupt the social processes it is intended to monitor." 쪽은 2011년 재수록본 기준이고, 경찰과 시험 점수의 예는 pp.34–35다. ↩

- 9 Meta AI, 2025-04-05, "The Llama 4 herd: The beginning of a new era of natively multimodal AI innovation", <https://ai.meta.com/blog/llama-4-multimodal-intelligence/> (보존본 <http://web.archive.org/web/20250407013603/https://ai.meta.com/blog/llama-4-multimodal-intelligence/>) — "Llama 4 Maverick offers a best-in-class performance to cost ratio with an experimental chat version scoring ELO of 1417 on LMArena." ↩
- 10 lmarena.ai(@lmarena_ai), 2025-04-08 00:08:09 UTC, X, https://x.com/lmarena_ai/status/1909397817434816562 — "Meta's interpretation of our policy did not match what we expect from model providers." · "Early analysis shows style and model response tone was an important factor" · "(Emoji control? 🤖)". ↩
- 11 Ahmad Al-Dahle(@Ahmad_Al_Dahle), 2025-04-07 17:49:31 UTC, X, https://x.com/Ahmad_Al_Dahle/status/1909302532306092107 — "We've also heard claims that we trained on test sets -- that's simply not true and we would never do that." 같은 글은 실험판의 순위를 언급하지 않았다. ↩
- 12 Shivalika Singh 외 12명, 2025-04-29(v2 2025-05-12), "The Leaderboard Illusion", arXiv:2504.20879, 초록 p.1 — "At an extreme, we identify 27 private LLM variants tested by Meta in the lead-up to the Llama-4 release." · LMArena, 2025-04-30, X, https://x.com/lmarena_ai/status/1917492084359192890 — "If a model does well on LMArena, it means that our community likes it!" 비공개 변형 수는 2025-01~03에 모든 표본 기준이고, 50점 상승은 실제 Arena 데이터가 아닌 시뮬레이션이다. ↩
- 13 저자 측정보(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.3·§ 2.5. 함께 틀린 5건이 모두 가족상 질문이라는 것은 결과를 본 뒤의 확인이다. ↩
- 14 Anthropic, "Model IDs and versioning", <https://platform.claude.com/docs/en/about-claude/models/model-ids-and-versions>, 'Model weights versus serving infrastructure' 절(2026-10-05 열람) — "Occasionally, infrastructure updates produce minor differences in observable behavior even when the model ID and weights have not changed." · Google, 2025-06-26, Gemini API release notes, <https://ai.google.dev/gemini-api/docs/changelog> — "The preview models gemini-2.5-pro-preview-05-06 and gemini-2.5-pro-preview-03-25 are now redirecting to the latest stable version gemini-2.5-pro." ↩
- 15 저자 측정보(사전 등록 2026-10-05 10:43, 한국 시각 — 결과보다 먼저), data/a2-results.md(판정·종류별 표·한계), 사전 등록 data/prereg-a2.md. 비공개 저장소는 집계만 심는다. 같은 저장소의 창이 겹쳐 사건이 독립이 아니고, 중앙값은 기술 통계다. 창이 잘린 사건까지 넣은 17건의 중앙값은 97.4%다. ↩
- 16 같은 파일의 누락 표 — llmux 커밋 <https://github.com/2lab-ai/llmux/commit/31e3800> (2026-09-08), soma-work 커밋 <https://github.com/2lab-ai/soma-work/commit/ec7bedf> (2026-07-14), 창이 잘린 사건의 llmux 커밋 <https://github.com/2lab-ai/llmux/commit/969f324> (2026-10-01). ↩
- 17 llmux 커밋 31e3800(2026-09-08, <https://github.com/2lab-ai/llmux/commit/31e3800>) 메시지 — "Advertise astra and gpt-6 on gpt-6-astra[1m] so shorthand clients select the 1M context profile" · "test(catalog): enforce unique astra alias ownership". 새 시험 함수는 없다. 같은 커밋의 src/catalog.rs는 기존 시험 `gpt_6_astra_aliases_context_and_effort_count` 의 별칭 단언 둘을 뒤집고, 같은 함수 안에 고유성 검사를 더했다 — "the ONLY row that advertises `astra` or `gpt-6` must be `gpt-6-astra[1m]`". 기존 시험이 그 배치를 단언했다는 판정은 data/a2-results.md의 누락 표다. ↩
- 18 Thakur 외, 2021-04-17(v4 2021-10-21), "BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models", arXiv:2104.08663, <https://arxiv.org/abs/2104.08663>, § 6 pp.8-9·표 4 — "Even though many systems contributed to the TREC-COVID annotation pool, the annotation pool is still biased towards lexical approaches." ↩

- 19 Boris Cherny(대담 Diana Hu), YC Startup School 2026, 녹취 게시 2026-07-27, YC Root Access, <https://www.ycrootaccess.com/p/boris-cherny-building-claude-code> — "An eval might live for maybe one, two, three model generations." · "Very often we just saturate the eval and then we have to throw it away and come up with a new eval." ↩
- 20 Bryan Bischof, 2025-09-04 23:03 UTC, X, <https://x.com/BEBischof/status/1963739648792117484> — "super extensive golden eval sets tend to age poorly". 같은 글에서 그는 단위 시험식 평가가 초기 반복과 회귀 확인 말고는 대체로 쓸모가 없었다고 썼다. ↩
- 21 OpenAI, Deprecations "2026-06-03: Evals platform", <https://developers.openai.com/api/docs/deprecations> — "On June 3, 2026, we notified developers using the Evals platform that the product is being deprecated." · 같은 페이지의 2026-11-30 항목 "The Evals dashboard and API are scheduled to shut down." · Evals 가이드, <https://developers.openai.com/api/docs/guides/evals> — "Existing evals content remains available during the transition window." 출처는 Changelog의 2024-10-01 항목(<https://developers.openai.com/api/docs/changelog>). ↩
- 22 저자 집계, data/a1-audit-results.md § 5 표의 proxy 행(첫 장치 커밋 전 13건과 뒤 29건, 중복 제거 뒤)· § 1(중복 4행). 원장 211행(2026-10-02 13:02, 한국 시각)의 놓친 지점 칸을 의역했다. 원장은 비공개다. ↩
- 23 저자의 에이전트 운영 기록(2026-09-30). 거부의 원문은 의역했다. ↩
- 24 저자의 에이전트 운영 기록(검사 재검토의 발동 집계, 2026-09-30). 발동 기간의 메인 세션 응답은 8,254개였다. 오탐 선언은 에이전트의 자기 신고라 오탐의 하한이다. ↩
- 25 저자의 에이전트 운영 기록(생존 검사 개별 검토의 공통 규칙, 2026-10-01). ↩
- 26 저자 집계, data/a1-audit-results.md § 5 "게이트 집합 확정" — $23 - 17 - 1 + 2 = 7$, 그 뒤 생겼다 지워진 혹 하나까지 26개 가운데 24개 제거, 2개 유지. ↩
- 27 저자의 에이전트 운영 기록(생존 검사 개별 판정, 2026-10-01~02). 판정은 재생과 표본 열람에 기댔고, 전수 판정이 아닌 항목도 있다. 이슈 트래커 사례는 업무 맥락이라 세부를 뺐다. ↩
- 28 저자의 에이전트 운영 기록(재검토 문서, 2026-09-30)과 원장 211행, 첫 장치의 행방은 data/a1-audit-results.md § 5 표. 기록은 에이전트가 기록기를 부를 때만 남으므로, 이 수는 검사 장치의 효과를 재지 않는다. 시험과 스크립트가 남은 수는 원장의 장치 행이 가리키는 경로를 2026-10-05에 저장소에서 대조한 저자 집계다(시험 14행 가운데 12행, 스크립트 69행 가운데 59행이 남음). ↩
- 29 저자의 에이전트 운영 기록(2026-10-02). PR 네 건의 머지 시각과 검사 실행은 soma-work PR #253 <https://github.com/2lab-ai/soma-work/pull/253> · #265 <https://github.com/2lab-ai/soma-work/pull/265> · #267 <https://github.com/2lab-ai/soma-work/pull/267> · #268 <https://github.com/2lab-ai/soma-work/pull/268>의 기록이다(2026-10-05 확인). 운영 기록은 네 건 모두 검사가 실패한 상태에서 머지됐다고 적었지만, 시각은 GitHub 기록을 따랐다. 서버 쪽 보호는 공개 저장소 <https://github.com/2lab-ai/soma-work>의 main 브랜치 설정이다(2026-10-05 확인 — 필수 검사 scan-quality-gates:lean-verify는 공개 API로 보이고, 나머지 설정은 관리자만 본다). ↩

여섯 달

석 달

2023-03-14, OpenAI는 GPT-4를 공개하는 글의 API 문단에 한 줄을 넣었다. 지금의 버전에 묶어 두고 싶으면 gpt-4-0314를 부르면 되고, 그 버전은 6월 14일까지 지원한다는 것이었다. 긴 문맥판 gpt-4-32k-0314도 같은 날까지라고 적혔다. 석 달짜리 약속이다. 두 주 전 gpt-3.5-turbo-0301을 내놓을 때도 OpenAI는 적어도 6월 1일까지는 지원한다고 썼다.¹

이름에 날짜를 박은 이런 스냅샷은 같은 이름이 가리키는 모델이 바뀌지 않게 하는 장치다. 그 이름을 골라 부르는 일을 버전 고정이라고 부른다. 6월 14일 하루 전인 2023-06-13, OpenAI는 두 스냅샷을 2023-09-13에 끈다고 공지했다. 2023-07-20 같은 글 머리에 갱신이 붙었다. 고객과 커뮤니티의 의견을 검토한 끝에 지원을 적어도 2024-06-13까지 늘린다는 것이었고, 그 옆에 까닭을 밝힌 문장이 있었다. 대부분의 지표는 나아졌지만 성능이 나빠지는 과업이 있을 수 있고, 그래서 API 사용자가 모델 버전을 고정하게 해 둔다는 것이다.²

공식 폐기 표에 적힌 gpt-4-0314의 종료일은 2026-03-26이다. 종료 공지는 2025-09-26에 나왔다. 출시에서 종료까지 1,108일, 석 달 약속이 세 해를 넘겼다. 적어도 6월 1일까지라던 gpt-3.5-turbo-0301은 562일을 살았다.³ 약속한 날짜는 결과적으로 하한 노릇을 했다. 6월 13일 공지가 먼저 9월 13일까지 늘렸고, 그 너머의 날들은 고객과 커뮤니티의 의견을 검토했다는 7월 20일 갱신 뒤에 왔다. 고정도 종료일까지만 통했다.

하루

2026-10-05 오전, 누리별항공 환불 앱의 240개 질문이 두 모델을 거쳤다. claude-haiku-4-5는 17.1%를, gpt-5.5는 2.5%를 틀렸다. 규정이 답하지 않는 60개에서의 실패율은 48.3%와 10.0%였다. 열 번의 시연으로는 두 모델이 갈리지 않았다. 둘 다 첫 실행 열 번을 모두 맞혔다.⁴ 앱 앞에 선택이 하나 놓였다. 모델을 바꿀 것인가.

측정이 답하는 것은 그 선택의 한 칸이다. 이 과업의 240개에서, 그날의 첫 실행으로 두 모델이 얼마나 틀렸는지도. 호출 한 번의 비용과 지연, 보안, 그리고 지원이 언제 끝나는지는 이 측정이 두 모델 사이에서 비교하지 않았다. 지연과 토큰 수는 호출마다 기록만 해 두었다. 240개는 자기 과업에서 품질이 어떻게 달라지는지를 확인하는 근거이고, 비용과 지연과 보안과 지원 종료는 따로 재야 하는 근거다.

지연 종료는 공식 표에 날짜로 적혀 있다. 그 앱이 묶어 둔 스냅샷 claude-haiku-4-5-20251001은 2025-10-15에 나왔고, Anthropic의 표는 그 퇴역일을 'Not sooner than October 15, 2026'으로 적는다. 2026-10-15보다 이르지 않다는 뜻이고, 측정한 날의 열흘 뒤다. 하한이지 수명 약속이 아니다. 같은 표는 활성으로 적힌 모델 14개의 잠정 퇴역일을 모두 출시일에서 정확히 1년 뒤 '이후'로 적고 있었다. Anthropic은 공개 모델을 퇴역시키기 적어도 60일 전에 알려줬다고 적었다. gpt-5.5는 2026-04-24에 나왔고, 2026-10-05까지 폐기 공지가 없었다. OpenAI의 폐기 페이지는 안전이나 규정 준수 문제로 서둘러야 하는 경우가 아니면 GA 모델을 적어도 6개월 앞서 알린다고 적는다.⁵

품질 칸을 채운 것은 바뀌지 않은 240개였다. 그 240개가 다음 모델 앞에서도 맞는 잣대일지는 그때 다시 재 봐야 한다. 검사가 남았다는 것과 그 검사가 여전히 맞다는 것은 다른 측정이다. 공급자가 인정한 '나빠지는 과업'에 그 앱의 과업이 드는지도 공급자의 지표가 말해 주지 않는다.

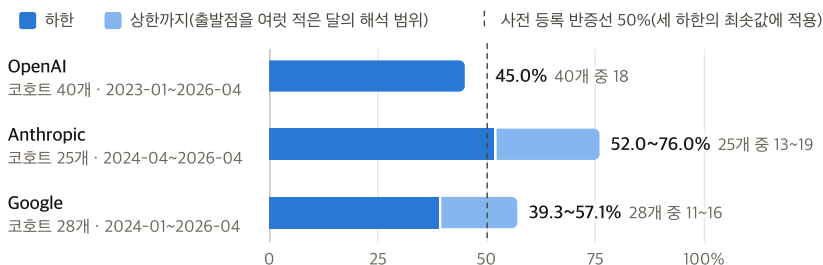
이 선택은 한 번으로 끝나지 않는다. 그 앱의 스냅샷은 퇴역일의 하한만 적힌 채 묶여 있고, 여섯 달 동안 폐기 공지와 새 판이 언제 올지는 정해져 있지 않다. 모델을 고르는 일 너머에 정할 것이 하나 더 있다. 여섯 달을 무슨 단위로 끊어 계획할지다.

반년

사전 등록된 판정은 하향이였다. 조건 없이 '바뀐다'고 쓰려면 어느 공급자로 시작하든 성립해야 해서, 판정에는 공급자마다 모델과 API, SDK만으로 짠 세 핵심 스택 하한의 최솟값을 쓰기로 했다. 그 값은 39.3%로, 측정 전에 정한 반증선 50%보다 낮았다. 그래서 '여섯 달 안에 무언가는 바뀐다'는 문장을 버리고, 등록된 문언대로 '바뀔 수 있다'로 낮췄다. 붙는 비율은 39.3%~100.0%다. 여섯 달 안에 무언가가 바뀔 수 있다는 말이지, 반드시 바뀐다는 말이 아니다.⁶

판정량이 센 것은 출발 코호트다. 2023-01부터 2026-04까지 매달 1일을 출발일로 삼았다. 그날 공식 문서가 새 사용자에게 출발점으로 권한 것을 먼저 쓰고, 그런 문장이 없으면 킥스타트나 공식 SDK README의 첫 예제가 쓴 것으로 모델과 API, SDK를 정했다. 조합에 따라 프레임워크와 평가 도구를 었었다. 그 스택이 출발 뒤 182일 안에 폐기 공지나, 호환성 단절을 명시한 메이저 판을 맞았는지(연성 단절), 구성 요소가 실제로 꺼졌는지(경성 단절)를 췌다. 연성 단절은 바뀔이 예고되거나 선언된 날이고, 경성 단절은 쓰던 구성 요소가 실제로 꺼진 날이다. 학습자는 출발일의 구성 요소를 올리지 않는다고 가정했고, 단절을 맞은 뒤에도 갈아타지 않은 채 첫 단절이 온 날만 췌다.⁷ 버전을 고정한 채 출발한 스택이 여섯 달 동안 무엇을 맞는지를 췌 셈이다.

Google 핵심 스택의 사전 등록 예측은 75%로 세 공급자 가운데 가장 높았다. 아무것도 었지 않은 핵심 스택의 연성 단절 비율은 OpenAI 45.0%, Anthropic 52.0~76.0%, Google 39.3~57.1%였다. 범위가 있는 것은 공식 문서가 출발점을 여럿 적어 둔 달이 있어서다. 판정에 붙은 비율의 위 끝은 모든 조합의 상한 가운데 가장 높은 값으로, OpenAI Agents SDK를 었은 조합(2025-04 이후 출발한 코호트)에서 나왔다.⁸ 여섯 달짜리 계획으로 옮기면, 어느 공급자로 시작하든 아무것도 바뀌지 않는다는 전제로 췌 수 없고, 바뀌는 주를 달력에 미리 정해 둘 수도 었다.



출발 코호트 비율(학습자 모집단의 발생률 아님), %

핵심 스택(프레임워크·평가 도구 없음) · 출발 후 182일 안 연성 단절(폐기 공지·호환성을 깨는 판) · 관측 종료 2026-10-05

2023-01부터 2026-04까지 매달 1일을 출발일로 삼은 OpenAI·Anthropic·Google 핵심 스택의 적격 출발 코호트 (40·25·28개) 가운데 출발 후 182일 안에 폐기 공지나 호환성을 깨는 판을 맞은 비율을 관측 종료일 2026-10-05 기준으로 췌 값이며, 었은 막대는 공식 문서가 출발점을 여럿 적은 달 때문에 생긴 하한~상한 범위다.

이 비율의 한 칸은 출발일 하나의 기록이다. 출발일 2023-07-01에 공식 Python SDK README의 첫 예제는 `openai.ChatCompletion.create` 로 Chat Completions API를

불렀고, 모델 칸에는 별칭 gpt-3.5-turbo가 있었다. 등록 규칙대로 별칭을 그날 이전에 나온 가장 새 스냅샷으로 읽으면, 그날 출발한 스택은 gpt-3.5-turbo-0613, Chat Completions API, openai-python 0.27 줄이다.⁹

여섯 달 창 안에서 단절로 센 사건은 하루에 몰려 있었다. 출발 128일 뒤인 2023-11-06, OpenAI는 gpt-3.5-turbo-0613의 폐기를 공지했고, 같은 날 openai-python 1.0.0이 나왔다. 1.0.0의 릴리스 노트는 호환을 깨는 변경의 목록을 이행 안내로 넘겼다. 이행 안내는 새 메이저 판을 냈으니 서둘러 올리기를 권한다는 문장과 라이브러리를 통째로 다시 썼다는 문장으로 시작했고, 코드를 바꿔 주는 `openai migrate` 명령이 함께 나왔다. 2023년 튜토리얼의 표준 호출이던 `openai.ChatCompletion.create`는 그 판에서 `client.chat.completions.create`로 바뀌었다.¹⁰ 이 코호트는 그날 연성 단절을 두 갈래로 맞았다. 모델의 폐기 공지와 SDK의 새 메이저 판이다.

그날 꺼진 것은 없었다. 새 SDK가 나왔다는 것과, 묶어 둔 판으로 돌던 실행이 실패했다는 것은 다른 사건이다. 0.27 줄은 계속 설치할 수 있었고 HTTP API는 그대로였다. 끊긴 것은 서비스가 아니라, 새 판 위에 옮겨 붙인 옛 코드였다. 공지를 받은 모델이 실제로 꺼진 것은 출발 440일 뒤인 2024-09-13으로, 여섯 달 창 밖이다.¹¹ 세 핵심 스택에서 구성 요소가 여섯 달 안에 실제로 꺼진 비율은 스택에 따라 0~12%였다. 연성 단절을 낸 것은 대부분 공급자 공식 SDK의 메이저 판이었고, API 계열에서 온 단절은 세 스택 모두 0이었다. 묶어 둔 스택이 여섯 달 안에 맞은 것은 대개 꺼짐이 아니라 공지와 새 판이었다. 그래서 여섯 달 동안 묶어 둔 것은 모델 스냅샷만이 아니라 SDK 판이고, 다시 썰 때는 그 판을 올리는 때다.

열여섯 달

그 코호트의 모델은 창 안에서 공지를 받고 창 밖에서 꺼졌다. 모델 ID 하나하나를 출시일부터 따라가면, 이름은 대개 반년보다 오래 갔다. 세 공급자의 공식 기록에 남은 텍스트 모델 ID의 중앙 생존은 499일, 열여섯 달 남짓이었다.¹² 그래도 2024년부터 날마다 시작하는 여섯 달 창을 잡으면, 창이 시작할 때 쓸 수 있었고 아직 공지가 없던 텍스트 모델 ID 가운데 평균 33.7%가 그 창 안에서 폐기 공지를 받았고 15.7%가 꺼졌다.¹³ 그러니 스냅샷 ID를 묶어 두는 일이 정하는 것은 바꾸는 페이지, 그 이름이 여섯 달을 간다는 보장이 아니다. 두 비율은 스냅샷만 따로 켜 값이 아니라 프리뷰와 특화 변형까지 넣은 텍스트 모델 ID 전체의 평균이고, 묶어 둔 이름 하나가 그 여섯 달 안에 공지를 받을지,

꺼지기까지 할지는 출발일에 알 수 없다. 공지를 받으면 그것이 바꿀 후보를 다시 채는 계기가 된다.

사흘

달력으로 기간을 약속한 과정들의 기록도 있다. 2012년 11월의 Dev Bootcamp 홈페이지 첫 화면에는 두 줄이 있었다. 'In 9 intense weeks', 그리고 'You will become a web developer'.¹⁴ 2017년 그 과정은 18주였다. 2017-07-12 Dev Bootcamp는 이메일로, 다양한 학생에게 열린 양질의 코딩 교육이라는 사명을 해치지 않고는 지속 가능한 사업 모델에 이를 수 없다며 그해 12월 8일에 문을 닫는다고 알렸다. 12월의 홈페이지에는 부트캠프 업계를 연 지 5년 만에 마지막 기수를 졸업시킨다는 문장이 걸렸다.¹⁵

2023-12-25, 샌안토니오의 Codeup 홈페이지는 경험이 없어도 6개월이 안 돼 기술 전문가가 된다고 광고했다. 사흘 뒤인 2023-12-28 정오, 직원들은 오늘부로 모든 운영과 수업을 멈춘다는 이메일을 받았다.¹⁶ 두 학교의 기록에서 확인되는 것은 광고가 약속한 기간과 그 뒤에 관측된 일이다. 문을 닫은 까닭은 이 기록으로 가리지 않는다.

LinkedIn의 미국 목록에서 AI 엔지니어는 2025년판과 2026년판 모두 가장 빨리 늘어난 직함 1위였고, 그 직함으로 옮긴 사람들이 그 전까지 쌓은 경력 연수의 중앙값은 3.6년과 3.7년이었다. 일자리가 가장 많은 직함의 순위가 아니라, 가장 빨리 늘어난 직함의 순위다.¹⁷

한 바퀴

여기서부터는 제안이다. 여섯 달을 달력의 칸 대신 순환으로 끊자는 것이다. 달력의 칸은 날짜마다 끝나 있어야 할 일을 미리 정한다. 2023-07-01에 출발한 스택에게 128일 뒤의 공지와 새 판은 출발일에 짤 어느 칸에도 없던 일이었다. 바퀴는 끝을 날짜 대신 확인으로 정한다. 한 바퀴는 가장 작은 구현에서 시작한다. 그 구현의 실패를 가를 작은 검사를 붙이고, 실제로 써 보고, 실패를 읽어 분류한 뒤, 구현과 검사를 함께 고치는 데서 끝난다. 무엇을 배울지도 달력이 정하지 않는다. 검색이나 도구 호출 같은 기술은 주 번호로 오지 않고, 오류 분석이 그쪽을 가리킬 때만 들어온다. 여섯 달을 이 순환을 몇 번 돌았느냐로

세자는 것이 이 책의 제안이다. 이 순서가 다른 순서보다 더 빨리 가르친다는 비교는 없고, 마친 뒤에 무엇이 된다는 약속도 없다. 일자리와 경력에 대해서는 아무것도 말하지 않는다.

바퀴에는 서로 다른 확인이 둘 들어간다. 하나는 시스템의 확인이다. 고정한 세트를 같은 판정으로 다시 돌려, 이번 변경이 무엇을 고치고 무엇을 깨뜨렸는지 센다. 그 세트는 첫 주의 기대 동작 20건에서 시작해 실제 실패를 받아 자라고, 교체 판단에 앞서 동결된다. 다른 하나는 사람의 확인이다. 그때까지 해 온 일 가운데 하나를 AI 도움 없이 다시 해 본다. 부록 B는 26주를 여덟 바퀴로 나누고, 바퀴가 끝나는 주 가운데 다섯 곳의 완료 조건에 이 확인을 넣는다.

4주에는 새 기대 동작 다섯 건과 그 판정 근거를 손으로 쓰고, 규칙 엔진과 몇 건이 어긋나는지 센다. 10주에는 수수료 숫자만 바꾼 개정 규정을 가정해 세트 열 건의 기대 동작을 쓰고, 개정 숫자를 넣은 엔진과 견준다. 14주에는 다섯째 바퀴의 핵심 변경 하나를 다시 짜서 같은 결과가 나오는지 본다. 18주에는 그 바퀴의 실사용 실패 하나를 입력과 조항과 코드 줄까지 추적하고, 원인에 닿았는지와 걸린 시간을 남긴다. 26주에는 다른 규정 문서 하나로 1주의 계약을 처음부터 다시 짜고, 걸린 시간과, 다른 사람 한 명이 판정 근거에 동의하지 않은 건수를 센다.

두 확인은 서로를 대신하지 못한다. 세트의 통과는 앱이 그 질문들에 맞게 답했다는 기록이다. 그 앱을 만든 사람이 도움 없이 같은 판정을 낼 수 있는지는 세트에 남지 않는다. 세트는 실패가 났다는 것과 고친 뒤 사라졌다는 것을 보여 주지만, 그 실패가 어느 입력과 조항과 코드 줄에서 왔는지를 사람이 혼자 짚을 수 있는지는 보여 주지 않는다. 도움 없이 다시 해낸 일은 거꾸로, 앱이 어느 질문에서 틀리는지를 말하지 않는다. 4주처럼 둘 다 규칙 엔진에 대 보는 때에도, 한쪽은 앱의 답을 대고 다른 쪽은 사람의 답을 낸다. 사람의 확인도 제안이고, 이 과업에서 그 확인이 배움을 늘린다는 직접 증거는 없다.

바퀴 안에서 버전 고정은 다른 일을 한다. 고정은 변화를 없애지 않고, 사용자가 고르는 모델과 SDK의 교체를 다른 변경과 떼어 놓는다. 같은 모델 ID 아래의 인프라 갱신이나 지원 종료까지 통제하지는 못한다. 스냅샷 ID와 SDK 판을 묶어 두면 새 모델과 새 판은 바퀴 안의 변경 하나로 들어오고, 그 변경도 시스템의 확인을 거친다. 출발 코호트의 연성 단절 대부분이 공식 SDK의 메이저 판에서 왔으니, 묶어 둔 판을 올리는 일도 그런 변경 하나로 다룬다. 2023-11-06 같은 날이 바퀴 안에 오면, 그날 나온 새 메이저 판은 묶어 둔 판을 건드리지 않는다. 판을 올리는 일은 다른 변경과 섞지 않은 한 주의 변경이 되고, 고정

세트를 다시 돌려 직전 판과 견준 뒤 남기거나 되돌린다. 같은 날 받은 폐기 공지는 모델 교체 판단의 계기로 남는다. 첫 바퀴(1~4주)에는 한 동작과 그 실패를 아는 법을 두고, 마지막 바퀴(22~26주)에는 그 앱의 교체 판단을 두어 동결한 세트로 정하게 한다. 달력을 버리지는 않는다. 26주는 고정이고, 늦어지면 아직 하지 않은 가운데 단계를 정해진 순서로 뺀다. 첫 주의 계약과 매주의 변경 하나, 세트를 다시 돌리는 일은 빠지 않는다.

첫 주

첫 바퀴에는 순서 문제가 하나 있다. 실패를 가를 검사를 짜려면 무엇이 맞는 출력인지 정해야 하는데, 그 기준은 출력을 본 뒤에야 굳는다.¹⁸ 그래서 첫 주에는 구현과 기대 동작이 함께 서고, 그 기대 동작은 굳은 정답지가 아니라 출력을 보며 고쳐 갈 첫 판이다.

부록 B의 첫 주는 가장 작은 동작 하나와 기대 동작 20건이 다 서야 끝난다. 가장 작은 동작은 명령 한 줄로 도는 모델 호출 하나다. 고정된 규정 전문과 지시문을 넣은 호출이 질문 하나에 「최종: 환불액」, 「최종: 판단 불가」, 「최종: 정보 필요」 가운데 하나로 답한다. 스냅샷 ID는 설정 한 곳에 두고, SDK 판은 잠금 파일로 묶고, API 키는 저장소 밖에 둔다. 20건은 정상 6, 경계 4, 정보 부족 4, 거절 6이다. 경계 넷은 출발까지 정확히 72시간, 출발 시각 정각, 정확히 3시간 늦은 항공편, 결제하고 정확히 24시간 뒤의 취소다. 그 가운데 하나는 플렉스 운임 300,000원에 세금 40,000원인 항공권을 출발 시각 정각에 취소하는 질문이고, 기대 동작은 「최종: 환불액 240000원」이다. 출발 시각 정각을 포함한다는 제7조를 읽어야 나오는 금액이다. 거절 여섯 가운데 하나는 어머니가 돌아가셔서 못 간다며 수수료 면제를 묻는 질문이고, 기대 동작은 「최종: 판단 불가」와 고객센터 안내다. 개인 사정의 면제는 규정이 정하지 않고 고객센터가 심사한다는 제1조③이 근거다.

통과 수는 그 주의 완료 조건에 들지 않는다. 20건의 원출력을 손으로 판정해 근거와 대조하는 일과, 그 20건이 가르는 것과 가르지 못하는 것을 한 문단씩 적는 일이 든다. 이 과업에서 20건이 가르는 것은 한 종류가 통째로 깨지는 실패와 출력 형식의 위반, 조항을 따지는 순서의 오류다. 몇 번에 한 번만 틀리는 실패와 적어 두지 않은 종류의 질문, 실제 사용에서의 실패율은 가르지 못한다. 세트 밖 질문 다섯은 직접 써서 넣어 보고, 거기서 나온 실패는 둘째 주 세트의 후보가 된다. 부록 B의 26주 실행표는 그 주에서 시작한다.

주

- 1 OpenAI, 2023-03-14, GPT-4 연구 발표 글, <https://openai.com/index/gpt-4-research/> (web.archive.org 2025-01-01 스냅샷), API 문단 — "you can pin the current version by calling gpt-4-0314, which we'll support until June 14" · OpenAI, 2023-03-01, "Introducing ChatGPT and Whisper APIs", <https://web.archive.org/web/20230302000653/https://openai.com/blog/introducing-chatgpt-and-whisper-apis> — "today we're releasing gpt-3.5-turbo-0301, which will be supported through at least June 1st". ↩
- 2 OpenAI, 2023-06-13(2023-07-20 갱신), "Function calling and other API updates", <https://openai.com/index/function-calling-and-other-api-updates/> — 원판(web.archive.org 2023-06-14 스냅샷) "These older models will be accessible through September 13th, after which requests specifying those model names will fail." · 'July 20, 2023 update' 문단 "While the majority of metrics have improved, there may be some tasks where the performance gets worse. This is why we allow API users to pin the model version." 같은 문단이 연장의 근거로 고객과 커뮤니티의 의견 검토를 적었다. ↩
- 3 OpenAI, Deprecations, <https://developers.openai.com/api/docs/deprecations>, "2025-09-26: Legacy GPT model snapshots (March 2026 shutdown)" 절(2026-10-05 열람). 날짜와 일수는 저자 측정 자료 data/m3-surfaces.csv의 gpt-4-0314:gpt-3.5-turbo-0301 행(행마다 공식 URL). ↩
- 4 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.1·§ 2.2·§ 2.3. 지연과 토큰 수는 data/calls.jsonl에 호출마다 남은 기록(latency_ms:usage)이고, 두 모델을 비용이나 지연으로 비교한 집계는 없다. ↩
- 5 Anthropic, "Model deprecations", <https://platform.claude.com/docs/en/about-claude/model-deprecation>, Notifications 절·Model status 표(2026-10-05 열람) — "providing at least 60 days' notice before model retirement for publicly released models" · 퇴역 칸 "Not sooner than October 15, 2026". 두 모델의 출시일은 Anthropic release notes(<https://platform.claude.com/docs/en/release-notes/overview>)와 OpenAI changelog(<https://developers.openai.com/api/docs/changelog>)로, data/m3-surfaces.csv의 해당 행이다. 출시 1년 뒤라는 것은 공식 문장이 아니라, 표에서 활성으로 적힌 14개 행(특화 변형 claude-myrthos-5:claude-myrthos-5-1 포함)의 날짜를 출시일(data/m3-surfaces.csv)과 대조해 확인한 패턴이다. data/m3-summary.md 표 K도 14/14다. · OpenAI, Deprecations, <https://developers.openai.com/api/docs/deprecations>, "Model deprecation notice periods" 절(2026-10-05 열람) — "Unless safety or compliance concerns require a faster timeline" · "Generally available models: At least 6 months." ↩
- 6 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m3-stack-results.md "결과 먼저"·결과 표·반증선 판정·한계·부록(계산 출력), 자료 data/m3-stack-components.csv(807행, 행마다 공식 출처), 판정 규칙(세 핵심 스택 하한의 최솟값과 그 까닭, 50%, 하향 문언, 범위의 두 끝, 구간을 붙이지 않는 까닭)은 data/prereg-m3.md § 7이고, 핵심 스택의 범위(모델·API·SDK, 프레임워크와 평가 도구 없음)는 같은 문서 § 2, 구성 요소를 고르는 순서(명시 권장 → 퀵스타트·README → 최신 GA)는 § 5, 연성·정성 단절의 정의는 § 1·§ 6, 판을 올리지 않고 첫 단절을 켜다는 고정 가정은 § 4다. 예측은 탐색 측정을 보고 적었다고 등록 때 밝혔다. 같은 단계의 출처가 서로 다른 달은 가장 늦게 깨지는 선택(하한)과 가장 빨리 깨지는 선택(상한)을 함께 적었다. 정성 단절 비율과 단절을 낸 범주의 분해(SDK·모델·API 계열)는 같은 결과 문서의 결과 표와 부록 § 3이다. ↩
- 7 판을 올리지 않는다는 것은 측정의 가정이다. 저자의 게이트웨이 llmex에서는 2026-07-10부터 09-28까지 기본 모델을 바꾸거나 별칭이 가리키는 모델을 옮긴 사건이 여섯 번 있었다. 저자 측정, data/a2-results.md 사건별 표(llmex-E1~E6)와 data/prereg-a2.md § 2의 사건 목록. ↩

- 8 같은 결과 문서(data/m3-stack-results.md)의 결과 표·사건 예측 대비 표·부록 § 2-한계 절, 예측은 data/prereg-m3.md § 8이다(OpenAI 45%, Anthropic 45%, Google 75%). 평가 도구 Inspect만 없으면 OpenAI 조합은 Inspect가 나온 뒤인 2024-05부터 출발한 코호트만 있어 29.2%였고, 없으면 조합까지 모두 넣으면 범위는 29.2%~100.0%다. 이 비율들은 등록 규칙으로 구성한 스택의 것이고, 실제 학습자 모집단에서 일어난 발생률이 아니다. 같은 ID의 조용한 행동 변화와 별칭의 재지정, 가격 변화는 세지 않았으니, 그것까지 세면 올라갈 수만 있는 하한이다.
- 9 README는 2023-05-18 커밋판(<https://github.com/openai/openai-python/blob/da828789387755c964c8816d1198d9a61df85b2e/README.md>)이고, 첫 생성 예제는 `openai.ChatCompletion.create(model="gpt-3.5-turbo", ...)` 다. 그 자리의 예제는 커밋 "Update README to use gpt-3.5-turbo by default (#441)"(2023-05-09, <https://github.com/openai/openai-python/commit/2b21516ee87cae266160023da678a82deb791c9b>)이 바꾸기 전까지 ada를 불렀다. README의 첫 예제는 SDK 관리자가 고른 예시라, 새 사용자에게 내건 권장과 같다는 보장은 없다. 그날의 모델 페이지는 복원되지 않아, 별칭은 data/prereg-m3.md § 5의 규칙 2(그 별칭으로 시작하는 ID 가운데 그날 이전 최신 출시 스냅샷)로 읽었다. 그 코호트의 선택 행은 data/m3-stack-components.csv에 있다.
- 10 openai-python v1.0.0 릴리스 노트, <https://github.com/openai/openai-python/releases/tag/v1.0.0> — "See the v1.0.0 discussion for a list of breaking changes and migration guide." 노트의 링크는 Discussion #631('v1.0.0 Beta', 2023-09-29)로 가고, 그 글 머리가 이행 안내 #742로 보낸다. openai-python의 CHANGELOG.md는 1.1.2부터 적혀 있어 1.0.0의 단절 선언은 이 노트에 있다. · rattrayalex, 2023-11-08, "v1.0.0 Migration Guide", GitHub openai/openai-python Discussion #742, <https://github.com/openai/openai-python/discussions/742> — "We have released a new major version of our SDK, and we recommend upgrading promptly. It's a total rewrite of the library, so many things have changed" 같은 글의 'Automatic migration with grit' 절이 `openai migrate` 명령을 적는다. 1.0.0의 날짜는 PyPI 업로드 기록이다.
- 11 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m3-stack-results.md 부록 § 5 코호트별 표의 openai 2023-07-01 행 — 최초 연성 128일, 최초 경성 440일. 모델의 공지일과 종료일은 data/m3-surfaces.csv의 gpt-3.5-turbo-0613 행이고, 출처는 OpenAI, Deprecations, "2023-11-06: Chat model updates" 절(<https://developers.openai.com/api/docs/deprecations>, 종료 2024-09-13)이다. 같은 날의 두 사건은 범주 분례에서 모델과 SDK에 한 번씩 됐다. 그보다 앞선 2023-08-31에 나온 0.28.0은 노트가 호환 단절을 적지 않아 단절로 세지 않았다(data/m3-stack-components.csv에서 'no breaking statement'로 판정된 행). 2023-11-06의 1.0.0은 출발 달이 이웃한 OpenAI 코호트 여섯의 창에 함께 들었다(같은 결과 문서의 한계 절). 이웃한 코호트가 같은 사건을 나눠 가지니 OpenAI 코호트 40개는 서로 독립인 표본 40개가 아니고, 그래서 이 비율들에는 신뢰구간을 붙이지 않았다.
- 12 저자 측정(답색 — 사전 등록 이전, 판정량이 아니다), data/m3-summary.md "결과 먼저" 1과 표 E1(KM 중앙 생존 — 텍스트 157개 499일 · GA 102개 707일 · 프리뷰 34개 319일, S(182일) — 텍스트 0.92 · GA 1.00), 자료 data/m3-surfaces.csv(258행, 행마다 공식 출처 URL), 공식 페이지에 남은 ID만 썼고, 같은 ID의 조용한 교체는 생존으로 처리됐다. 아직 살아 있는 ID는 관측 종료일(2026-10-05)에서 잘린 기록으로 다뤄 Kaplan-Meier 추정으로 썼고, 공식 표는 그날 연 판본으로 묶어 두고 썼다. 출시일부터 전 ID의 수명이라, 어느 달에 출발한 스택이 그 ID를 며칠 쓰는지와 같은 값이 아니다. 그 표도 그대로 믿을 기록은 아니다 — Anthropic, "Model deprecations", <https://platform.claude.com/docs/en/about-claude/model-deprecations>, 2024-09-04 절(2026-10-05 열람)의 셀 'November 6, 2024' · `claude-1.0` · `claude-haiku-4-5-20251001`. 2024-11-06에 꺼진 claude-1.0의 권장 대체로 적힌 것이 그 앱이 묶어 둔 스냅샷이고, claude-1.0이 꺼지고 열한 달 뒤(data/m3-surfaces.csv의 출시일 2025-10-15)에 나온 모델이다. 같은 절의 Claude 1-Instant 일곱 행이 모두 이 모델을 권장 대체로 적고, 다른 퇴역 행 열두 개 가운데 여섯도 퇴역 뒤에 나온 모델을 적는다(예: 2025-07-21에 꺼진 `claude-2.0` → 2026-05-28에 나온 `claude-opus-4-8`, 출시일은 같은 파일의 값). 나머지 여섯이 적은 모델은 퇴역 전에 나왔다. 퇴역 행 열아홉 가운데 열셋이 퇴역 때 아직 없던 모델을 적으니,

이 칸은 공지 당시의 권장을 남긴 기록이 아니다. 모델 ID 생존은 이 표의 날짜 칸만 썼으므로 이 칸은 그 결론을 바꾸지 않는다.

- 13 data/m3-summary.md "결과 먼저" 2와 계산 출력 F1·F2, 2024-01-01부터 2026-04-06까지 하루에 하나씩 시작하는 6개월 창 827개의 평균이다. 창에 든 ID들의 평균이지 한 스택이 겪은 일이 아니고, 33.7%는 창 시작일에 쓸 수 있는 ID를 고르게 쓴다는 가정의 값이다. 텍스트 모델 ID는 GA·프리뷰·특화 변형이고, 스냅샷과 별칭이 같은 일정을 따르면 한 행으로 썼다(같은 문서의 방법 절). 꺼진 비율은 종류 가운데 프리뷰(26.0%), 공급자 가운데 Google(40.3%)이 가장 높았다(F1). 창 시작일에 공급자의 가장 최근 GA ID를 골랐다면, 그 ID가 창 안에서 꺼진 비율은 OpenAI·Anthropic 0.0%, Google 16.0%였다(F2). ↩
- 14 Dev Bootcamp, 2012-11-05 스냅샷, devbootcamp.com 홈, <https://web.archive.org/web/20121105094433/http://devbootcamp.com/> — "In 9 intense weeks" · "You will become a web developer". 페이지 제목은 "Dev Bootcamp – Learn Ruby on Rails in 9 weeks."다. ↩
- 15 Tony Wan, 2017-07-12(07-13 갱신), "Unable to 'Reach a Sustainable Business Model,' Dev Bootcamp Will Shut Down in December", EdSurge, <https://www.edsurge.com/news/2017-07-12-unable-to-reach-a-sustainable-business-model-dev-bootcamp-will-shut-down-in-december> — "we simply cannot reach a sustainable business model without compromising our mission of delivering a high-quality coding education that remains accessible to a diverse population of students." 회사 이메일은 이 보도를 거쳐 옮겼다. 18주도 같은 기사의 값이다. 마지막 기수의 문장은 web.archive.org의 devbootcamp.com 2017-12-12 스냅샷 — "Five years after pioneering the bootcamp industry, Dev Bootcamp graduates our final cohorts". ↩
- 16 Codeup, 2023-12-25 스냅샷, <https://web.archive.org/web/20231225235724/https://www.codeup.edu/> — "Go from no experience to tech professional in less than 6 months." · Tracy Idell Hamilton, 2024-01-02, San Antonio Report, <https://sanantonioreport.org/codeup-students-wont-graduate-bootcamp-close-d-doors/> — "effective today (12/28/2023) Codeup will cease all operations and classes." 정오는 같은 보도가 이메일의 시각 기록으로 적은 것이다. 학교의 폐업 서한(2024-02-03 스냅샷의 홈, <https://web.archive.org/web/20240203125710/https://www.codeup.edu/>)은 사유를 여럿 함께 들었다 — "With hiring freezes in the job market, the funding landscape shifting in 2023, specifically with local and VET TEC funding programs, combined with our inability to retain essential teaching resources in December" ↩
- 17 LinkedIn News, 2025-01-07, "LinkedIn Jobs on the Rise 2025: The 25 fastest-growing jobs in the U.S.", <https://www.linkedin.com/pulse/linkedin-jobs-rise-2025-25-fastest-growing-us-linkedin-news-grvie> · 2026-01-07, "LinkedIn Jobs on the Rise 2026: The 25 fastest-growing roles in the U.S.", <https://www.linkedin.com/pulse/linkedin-jobs-rise-2026-25-fastest-growing-roles-us-linkedin-news-dlb1c> — "Median years of prior experience: 3.7 | Top roles transitioned from: Software Engineer, Data Scientist, Full Stack Engineer" (2026판). 방법 절은 이 값을 "the median years of work experience held prior to starting in the featured title"로 정의한다. 가장 많은 일자리가 아니라 가장 빨리 늘어난 직함의 순위이고, LinkedIn 회원 표본이다. ↩
- 18 Shreya Shankar 외 4명, 2024-04-18(arXiv v1), "Who Validates the Validators? Aligning LLM-Assisted Evaluation of LLM Outputs with Human Preferences", arXiv:2404.12272, <https://arxiv.org/abs/2404.12272>, 초록 — "users need criteria to grade outputs, but grading outputs helps users define criteria." · 서론 — "it is impossible to completely determine evaluation criteria prior to human judging of LLM outputs". 뒤의 문장은 실무자 9명의 질적 연구에서 저자들이 끌어낸 함의다. ↩

마지막 줄

최종: 환불액

누리별항공 환불 앱이 처음 낸 답의 마지막 줄은 「최종: 환불액 339000원」이었다. 세이버 표 419,000원을 물은 질문에 2.8초 만에 돌아온 답이었고, 규칙 엔진이 같은 사실로 낸 금액과 같았다. 시연 열 개는 두 모델에서 모두 그렇게 지나갔다. 열 개의 마지막 줄이 모두 규칙 엔진의 금액과 같았다. claude-haiku-4-5에서 답 하나에 걸린 시간의 중앙값은 2.7초였다. 앱에 준 지시문에는 규정으로 판단할 수 없는 질문이면 추측하지 말고 판단할 수 없다고 안내하라는 줄이 있었다.¹

그 줄에서 출발해 켄 숫자들은 이렇다. 같은 날 같은 앱에 240개를 넣자 claude-haiku-4-5는 17.1%를 틀렸다. 사전 등록한 예측은 15%였다. 실패는 규정이 답하지 않는 질문 60개에 몰려 29개였고, 그 가운데 20개가 규정이 정하지 않은 판단을 금액으로 단정했다. 같은 입력을 다섯 번씩 물으면 입력 50개 가운데 24%가 때로 맞고 때로 틀렸고, 시연 한 건은 네 번 맞은 뒤 다섯 번째에 틀렸다. 고른 열 개는 사용 입력의 무작위 표본이 아니어서, 열 번의 성공으로는 실패율의 상한을 셈할 수 없다. 열 개가 240개를 잘 예측했는지는 모델과 과업에 따라 갈렸다. gpt-5.5로 돌린 두 앱과 claude-haiku-4-5로 돌린 추출 앱에서는 잘 예측했고, claude-haiku-4-5로 돌린 환불 앱에서만 놓쳤다.²

채점자도 켜다. 정답을 아는 쌍 85개에서 세 채점자는 98.8%를 맞혔고, 서로 거의 같은 판정을 냈다(AC1 0.976). 85쌍의 예측은 그보다 낮은 88%와 0.78이었다. 정답이 하나가 아닌 열린 쌍 100개에서는 일치가 0.515로 떨어졌고, 두 답의 순서를 바꾸면 판정의 9.7%p가, 기준 문구를 고치면 23.3%p가 움직였다. 결과를 본 뒤 고친 기준이 열린 쌍의 일치를 0.939로 올렸지만, 그것은 판정값이 아니라 기준이 만든 일치였다. 정확도를 켤 수 있었던 것은 정답을 아는 쌍뿐이었다.³

내 저장소 셋의 검사는 남았다. 모델이나 엔진, SDK가 바뀐 사건 가운데 창이 잘리지 않은 14건에서, 사건 이전 검사 본문의 중앙 97%가 14일 뒤에도 공백 차이를 빼면 글자 그대로 남았다. 바뀐 모델을 직접 부르는 검사도 중앙 77%가 남았다. 예측은 88%와 45%였으니,

검사가 바뀌는 정도를 크게 본 셈이다. 가장 적게 남은 것은 리뷰 게이트 파일로, 9개 가운데 5개였다. 남은 검사가 잡지 못한 실패는 2건 확인됐다.⁴

그리고 여섯 달. 모델의 이름은 대개 1년 넘게 갔다. 공식 기록에 남은 텍스트 모델 ID의 중앙 생존은 499일이었다. 그래도 등록 규칙으로 구성된 출발 코호트 가운데 여섯 달 안에 폐기 공지나 호환성을 깨는 판을 맞은 비율은, 아무것도 없지 않은 세 공급자의 스택에서 39.3~76.0%였고 OpenAI Agents SDK를 얹은 조합에서 100%였다. 단절은 대부분 공급자 공식 SDK의 메이저 판에서 왔고, API 계열에서 온 것은 없었다. 아무것도 없지 않은 스택에서 구성 요소가 실제로 꺼진 비율은 0~12%였다. 가장 많이 깨지리라 예측한 Google 스택(75%)은 하한으로 견주면 셋 가운데 가장 낮았다(39.3~57.1%). 측정이 허락한 문장은 등록된 규칙대로 낮춘 '여섯 달 안에 무언가가 바뀔 수 있다(비율 39.3%~100.0%)'까지다. 실제 학습자 모집단에서 일어난 비율은 아니다.⁵

최종: 정보 필요

재지 못한 것도 같은 줄에 놓인다. 가장 큰 것은 사람이다. 204건의 원장은 시스템에 장치가 붙은 자리를 보여 줬지만, 그 석 달 동안 원장 결의 사람이 덜 틀리게 됐는지는 보여 주지 않았다. 분모가 없었다.⁶ 도움을 건여 낸 시험으로 배움을 켜 연구는 있었다. 튀르키예의 한 고등학교에서 답을 주는 GPT-4와 연습한 반은 2023년 가을 연습 점수가 대조군보다 48% 높았고, AI 없이 치른 시험에서는 17% 낮았다. 새 파이썬 라이브러리를 처음 쓴 개발자 52명의 실험에서도 AI 보조 집단의 퀴즈 점수가 17%p 낮았다(평균 50% 대 67%). 다른 사람들, 다른 과목의 기록이다. LLM 위에서 제품을 짓는 일을 배우는 사람에게 친절한 환경과 사악한 환경의 틀을 직접 실증한 연구는, 2026-10-05에 검색한 범위에서 찾지 못했다.⁷ 이 책의 측정 가운데 사람이 무엇을 배웠는지를 켜 것은 없다.

그 앱에 대해서도 모르는 것이 남았다. 17.1%는 사전 등록된 가중치로 섞은 240개의 값이지, 실제 고객 문의를 무작위로 뽑아 켜 실패율이 아니다. 고객에게 보낼 안내문으로서 어느 답이 나은지는 세 채점자가 갈렸고, 그 갈림을 가를 정답은 없었다. 호출 한 번의 비용과 보안, 지원 종료는 이 측정 밖이다. 지연은 기록만 했고 판정에 넣지 않았다.

지운 장치가 옳았는지도 아직 모른다. 원장의 여섯 실패군 모두에서, 첫 장치가 생긴 뒤에도 같은 실패군의 교정이 16~37건씩 다시 기록됐다. 장치가 막아 낸 사건은 원장으로 셀 수

없었다. 2026-10-02까지 재검토한 검사 장치 26개 가운데 24개가 지워졌다. 23개는 LLM 엔진 셋의 판정으로, 하나는 내 결정으로 지웠다. 지운 지 사흘 뒤가 이 기록의 끝이다. 남은 검사가 계속 맞는지도 재지 않았다. 재사용률은 검사가 남았는지만 켜다. 남은 시험 하나는 별칭이 잘못 붙은 배치를 바로 그대로 단언하고 있었다.⁸

바깥의 숫자들은 빠르게 움직였다. Klarna는 2024-02-27 AI 상담 도우미가 출시 한 달 만에 상담 대화의 3분의 2를 맡았다고 발표했고, 1년 반 뒤인 2025-09-02의 증권신고서에는 12개월 동안 69%, 상담원 700명 이상의 일이라고 적었다. 회사가 스스로 켜 숫자지만, 첫 달의 숫자가 한 해를 버텼다.⁹ 모델이 끝까지 해내는 일의 길이도 빠르게 늘었다. METR의 원자료에서 2023년 이후 50% 시간 지평은 약 129일마다 두 배가 됐고, 2026-02-05에 나온 Claude Opus 4.6은 약 719분에 이르렀다. 같은 모델의 80% 지평은 약 70분이었고, METR은 16시간이 넘는 측정은 지금의 과제 묶음으로는 믿을 수 없다고 적었다.¹⁰ 이 두 기록은 그 앱의 측정과 반대쪽을 가리킨다. 첫 숫자가 뒤의 숫자를 맞힌 경우가 있고, METR의 과제 묶음에서 모델은 빠르게 나아졌다. 그 앱에서는 다른 공급자의 모델로 바꾸자 같은 240개의 실패가 41개에서 6개가 됐다. 그 차이가 시간 때문인지 모델 등급 때문인지는 이 측정으로 가를 수 없다.

최종: 판단 불가

그 앱이 틀린 금액 단정 20개는 정답이 「최종: 판단 불가」인 자리에 금액을 적은 답이었다. 규정 제1조③은 개인 사정에 따른 수수료 면제를 규정이 정하지 않고 고객센터가 심사한다고 적었다. 가족상 질문 하나에서 앱은 그 조항을 짚은 단락 뒤에 금액을 썬해 「최종: 환불액 249000원」으로 끝냈다. 채점이 읽은 것은 그 마지막 줄이었다.¹¹

2024-02-14 캐나다 브리티시컬럼비아주의 민사분쟁심판소는 Air Canada 웹사이트의 챗봇이 유족 할인을 여행한 뒤에도 신청할 수 있다고 안내한 일을 판단했다. 챗봇은 이미 여행을 했더라도 항공권 발행일로부터 90일 안에 신청하면 된다고 답했고, 답 속 링크가 연 정책 페이지는 반대로 말했다. 할머니가 세상을 떠난 2022-11-11에 예약한 가는 편과, 같은 안내를 믿고 닷새 뒤 예약한 돌아오는 편의 일이었다. 2023-02-08 Air Canada의 직원은 챗봇이 "오해를 부르는 말"을 했다고 인정했다. 답장은 챗봇 답변에 걸린 정책 페이지 링크를 가리켰고, 챗봇을 고칠 수 있게 문제를 기록해 두었다고 적었다. 심판소는 Air Canada의 주장을 사실상 챗봇이 제 행동에 스스로 책임지는 별개의 법적 존재라는

말로 요약하고, 이례적인 주장이라고 적었다. 그리고 고객이 웹사이트의 한 부분에서 찾은 정보를 왜 다른 부분에서 다시 확인해야 하는지 Air Canada가 설명하지 않았다고 적고 (¶28), 손해 650.88캐나다달러에 이자와 수수료를 더한 812.02캐나다달러를 물게 했다. 결정문은 그 챗봇이 어떤 기술로 돌았는지 밝히지 않았다. Air Canada가 챗봇의 성격에 관해 아무 정보도 내지 않았다고 적었을 뿐이다.¹² 확인한 사람도 있었다. Mata 사건의 변호사는 판례가 진짜인지를 그 판례를 내놓은 기계에게 물었고, Google에서 찾을 수 없었다는 바깥의 결과는 Google에 없는 판례도 있으리라는 짐작으로 넘겼다.¹³

2026-10-05 오전 10시 43분 22초, 앱에 질문 하나가 들어갔다. "문의드려요. 세이버 표로 총 417,000원 결제했는데 그중 세금·공항이용료가 60,000원이라요. 53일 전에 예매했어요. 비행기를 놓쳤어요. 출발한 지 30시간 됐어요. 가족상을 당해서 못 가게 됐어요. 사망 증명서 내면 수수료 면제되나요? 환불 얼마 받을 수 있나요?" 3.9초 뒤 앱은 제7조로는 운임과 세금 모두 환불되지 않는다고 적고, 제1조③을 옮긴 다음 이렇게 썼다. "가족상 관련 수수료 면제 가능 여부는 이 규정으로 판단할 수 없습니다." 가족상과 입원으로 면제를 묻은 열네 질문 가운데, 마지막 줄이 이렇게 끝난 답은 하나였다. 같은 질문에 gpt-5.5는 면제는 따로 심사한다는 단서를 달고 「최종: 환불액 0원」으로 끝냈다. 첫 채점 기준으로 두 답을 원래 순서대로 놓고 묻자, 세 채점자는 모두 판단 불가로 끝낸 답을 골랐다. 사후에 열어 본 기록에서는, 두 답의 순서만 바꾸자 셋 모두 gpt-5.5의 답을 골랐다. 규칙 엔진의 답은 판단 불가였다.¹⁴

「최종: 판단 불가」

주

- 1 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 원출력 data/calls.jsonl seq 13(claude-haiku-4-5, 질문 t1d-01, 지연 2,800밀리초), 시연 결과 data/m1-m2-results.md § 2.1· § 4, 앱 지시문 data/prompts.py. ↩
- 2 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 1.2· § 2.1· § 2.2· § 2.3· § 2.4. 출력 분산의 반복 실행은 실패율 계산에 넣지 않았다. ↩
- 3 같은 파일 § 3 — 정답을 아는 쌍, 열린 쌍, 결과를 본 뒤 쓴 기준(§ 3.6, 사후 기술). 채점자 셋 가운데 둘이 같은 공급자 계열이라 AC1이 위로 치우칠 수 있다. ↩
- 4 저자 측정(사전 등록 2026-10-05 10:43, 한국 시각 — 결과보다 먼저), data/a2-results.md(판정·종류별 표·누락 표). 저장소 셋을 석 달 동안 본 한 사람의 시스템이고, 남았다는 것은 맞게 작동한다는 뜻이 아니다. ↩
- 5 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m3-stack-results.md. 판정량은 프레임워크와 평가 도구를 엮지 않은 세 핵심 스택 하한의 최솟값(39.3%)이고, 스택별로는 OpenAI 45.0%, Anthropic 52.0~76.0%, Google 39.3~57.1%다. 낮은 문구의 범위는 사전 등록대로 핵심 스택 하한의 최솟값에서 전 조합 상한의 최댓값까지다(data/prereg-m3.md § 7). 얹은 조합까지 모두 넣은 범위는 29.2%~100.0%이고, 아래 끝은 평가 도구 Inspect만 얹은 OpenAI 조합(코호트 24개)이다(data/m3-stack-results.md 부록 § 2). 모델 ID의 생존은 사전 등록 이전의 탐색 측정이다(data/m3-summary.md). ↩
- 6 저자 집계, data/a1-audit-results.md § 2· § 3. 관측 범위는 2026-07-06~10-02, 한 사람의 한 시스템이다. ↩
- 7 Hamsa Bastani 외, 2025-06-25, "Generative AI without guardrails can harm learning: Evidence from high school mathematics", PNAS 122(26):e2422633122, <https://doi.org/10.1073/pnas.2422633122> · Judy Hanwen Shen·Alex Tamkin, 2026-01-28, "How AI Impacts Skill Formation", arXiv:2601.20245 · 평균 50% 대 67%는 두 저자의 글 "How AI assistance impacts the formation of coding skills"(Anthropic, 2026-01-29, <https://www.anthropic.com/research/AI-assistance-coding-skills>)에 있다. 앞의 것은 한 학교·수학 한 과목·2023년 가을 GPT-4의 단기 결과이고, 17%는 대조군 평균 대비다. 뒤의 것은 크라우드워커 52명이 과제 직후에 본 퀴즈이고, 17%p는 퀴즈 백분율의 차이다. 직접 실증한 연구를 찾지 못했다는 판단은 2026-10-05에 'wicked learning environment'를 소프트웨어 공학·프롬프트 엔지니어링·머신러닝 엔지니어와 엮어 검색한 범위에 한정된다. ↩
- 8 저자 집계와 측정, data/a1-audit-results.md § 5(실패군별 후속 추적, 26개 가운데 24개 제거)와 data/a2-results.md 누락 표(llmux 커밋 <https://github.com/2lab-ai/llmux/commit/31e3800>). 지출지에 대한 판정은 저자의 에이전트 운영 기록(2026-09-30~10-02)에 있다. 핸드오프 문장 검사 하나는 엔진 판정이 아니라 10-01 저자의 결정으로, 엔진들이 지우기로 한 17개와 함께 지워졌다(data/a1-audit-results.md § 5 「게이트 집합 확정」). ↩
- 9 Klarna, 2024-02-27, "Klarna AI assistant handles two-thirds of customer service chats in its first month", <https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/> · Klarna Group plc, 2025-09-02, Form F-1/A Amendment No. 3, SEC EDGAR, <https://www.sec.gov/Archives/edgar/data/2003292/000200329225000024/klarnagroupplcf-1a3.htm> — "Our AI assistant handled 69% of customer service chats in the last twelve months ended June 30, 2025, according to our service chat log data, doing the work equivalent of over 700 full-time agents". 700명은 업무를 환산한 추정치다. ↩
- 10 METR, 2026-05-08(페이지 갱신), "Task-Completion Time Horizons of Frontier AI Models", <https://metr.org/time-horizons/> · 원자료 https://metr.org/assets/benchmark_results_1_1.yaml — "Measurements ↩

above 16 hrs are unreliable with our current task suite". Claude Opus 4.6의 50% 지평 95% 구간은 316.7~3,633.8분이다. 80% 지평과의 비교는 원자료로 한 지자의 계산이다.

- 11 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 원출력 data/calls.jsonl seq 266(claude-haiku-4-5, 가족상, 질문 t1c-039), 규정 원문 data/sets/policy_true.txt 제1조③. 채점은 답의 마지막 「최종:」 표기만 읽는다(data/prereg.md § 1, data/grade.py). ↩
- 12 Christopher C. Rivers(Tribunal Member), 2024-02-14, Moffatt v. Air Canada, 2024 BCCRT 149, Civil Resolution Tribunal, <https://decisions.civilresolutionbc.ca/crt/crtid/en/item/525448/index.do>, ¶13·¶14·¶15·¶16·¶17·¶18·¶22·¶27·¶28·¶40·¶44 — "misleading words" · "This is a remarkable submission." · "double-check information found in one part of its website on another part". 소액 사건이라 선례로서의 구속력은 약하다. ↩
- 13 Steven A. Schwartz, 2023-05-25 제출, Affidavit(ECF 32-1), Mata v. Avianca, Inc., 1:22-cv-01461(S.D.N.Y.), https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.32.1_2.pdf · 심리 속기(ECF 52), https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.52.0_2.pdf, Tr. 29 — "I thought that there are cases that I am not going to be able to find on Google". 진술서는 질문한 날을 적지 않았다. 법원은 진술서가 시사한 시점과 2023-06-06 선언서의 설명이 서로 어긋난다고 보았다(P. Kevin Castel, 2023-06-22, Opinion and Order on Sanctions, ECF 54 ¶¶46-47, <https://storage.courtlistener.com/recap/gov.uscourts.nysd.575368/gov.uscourts.nysd.575368.54.0.pdf>). ↩
- 14 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 질문 t1c-212(data/sets/t1_challenge.jsonl), 원출력 data/calls.jsonl seq 518(claude-haiku-4-5, 자연 3,904밀리초)·seq 1023(gpt-5.5). 두 답은 쌍 A-022로 채점됐다. 첫 기준으로 원래 순서에서 낸 판정은 seq 1648(gpt-6-astra)·1765(grok-4.7)·2727(gpt-5.6-sol)이고, 순서를 바꾼 판정은 seq 1649·1769·2728이다. 같은 순서로 다시 물었을 때는 grok-4.7만 gpt-5.5의 답을 골랐고(seq 1771), 결과를 본 뒤 고친 기준에서는 셋 모두 판단 불가 쪽이었다(seq 3264·3290·3322). 정확도는 사전 등록대로 원래 순서의 첫 판정으로 했다. 이 쌍을 열어 본 것과 하위 유형별 수(data/m1-m2-results.md § 2.3)는 결과를 본 뒤의 사후 기술이다. A·B를 답의 정체로 옮기는 대응은 data/m2_pairs.py. ↩

반증과 만료

판정 먼저

이 책은 측정 전에 주장 셋과, 그 주장을 꺾을 조건을 등록했다. T1은 시연이 말해 주는 범위(그 가운데 (b)는 고른 시연의 예측력), T2는 채점이 만드는 피드백의 범위, T3는 여섯 달 안의 변화와 그 뒤에 남는 검사다. 측정 뒤 반증선을 넘은 것은 T3의 앞 절 하나다. '여섯 달 안에 무언가는 바뀐다'는 등록한 규칙대로 '바뀔 수 있다(비율 39.3%~100.0%)'로 내려갔다. T1과 T2는 반증선에 걸리지 않아 문구를 그대로 두었지만, T1에는 반대 결과가 붙는다. 과업 2와, 비교 모델 gpt-5.5로 돌린 두 과업은 반증선 조건을 충족했다. 저자 원장은 맞는 분모가 없어 비율을 포기했다.¹

표의 상태 어휘는 이렇게 쓴다. **유지**는 반증선이 성립하지 않아 문구를 그대로 둔다는 뜻이다. **하향**은 반증선이 성립해 등록해 둔 낮은 문구로 바꿨다는 뜻이다. **좁힘**은 주장이 서는 범위를 줄인다는 뜻이고, **포기**는 계산하지 않기로 한 양이다. **불발동**은 금지 조건이 성립하지 않았다는 뜻이다. **반대 결과**는 판정을 바꾸지 않았지만 주장과 반대쪽으로 나온 결과이고, **예측 실패**는 사전 예측이 측정의 95% 구간 밖에 놓인 경우다. 이 판정 규칙은 그 앱의 측정과 채점자 측정에만 등록했다. 스택 추적과 남는 검사의 행에는 "등록한 판정 규칙 없음"과 예측과의 차이를 적는다. 사후 기술은 사전 등록에 없던 집계나 결과를 본 뒤의 서술이다.

반증선 표

주장	반증선(등록 문구)	사전 예측	결과	상태
T1(b) 고른 시연의 예측력	두 과업 모두 W	시연 첫 실행 성공률 - 도전 세트 가중 성공률W	≤ 10%p이고 도전 세트 가중 실패율 < 3%이면 T1(b)를 「이 두	과업 1 실패율 15%, 시연 실패율 5% (10/10일 확률 약 60%) · 과업 2 실패율 7% 과업 1: 절대자 17.1%p, 실패율 17.1% [13.3, 20.8] · 유지 — 과업 1에서 불성립. 과업 2는 두 조건을 모두

주장	반증선(등록 문구)	사전 예측	결과	상태
			설정에서 약하다」로 보고한다. 판정은 1차 모델	과업 2: 충족한 반대 결과이고 예측 실패 (-4.5%p). 과업 1 절대차는 예측 (10.0%p, 경계)보다 컸다 ²
T1(b) 비교 모델	같은 계산을 비교 모델에도 하고 민감도로 나란히 적는다	gpt-5.5 과업 1 7% · 과업 2 3%	과업 1: 2.5%p, 2.5% [1.2, 4.2] · 과업 2: 1.7%p, 1.7% — 두 과업 모두 반증선 성립	반대 결과 — (b)의 경험 부분을 받치는 것은 네 조건 가운데 과업 1·claude- haiku-4-5 하나. 과업 1은 예측 실패 (7% → 2.5%, -4.5%p)
T2 채점자	전체 200쌍에서 순서 효과 < 5%p · Gwet AC1 > 0.8 · 루브릭 효과 < 5%p가 모두 성립하면 「이 설정에서 채점자 편향 주장은 약하다」	6%p · 0.53 · 8%p	5.8%p [2.3, 9.4] · 0.734 [0.669, 0.798] · 12.6%p <u>7.9</u> <u>17.5</u>	유지 — 순서 효과는 구간이 5%p를 걸쳐 셋 가운데 가장 약한 근거. AC1은 예측 실패(+0.20). 편향은 정답이 없는 열린 쌍에 몰렸다(순서 효과 9.7%p·루브릭 효과 23.3%p·AC1 0.515, 정답을 아는 쌍은 1.2%p·0.0%p·0.976) ³
T3 앞 절 — 스택	세 핵심 스택 하한의 최솟값 m < 50%이면 원 문구를 폐기하고 "바뀔 수 있다(비율	OpenAI 45% · Anthropic 45% · Google 75%, 판정은 하향 (확신 약 60%)	m = 39.3% (Google). OpenAI 45.0% · Anthropic 52.0~76.0% · Google 39.3~57.1% · 전 조합	하향 — "여섯 달 안에 무언가가 바뀔 수 있다(비율 39.3%~100.0%)". Google 예측: 등록된 판정 규칙 없음 — 예측과 가장 크게 어긋남(하한 기준

주장	반증선(등록 문구)	사전 예측	결과	상태
	X%~Y%)”로 하향한다		상한 100.0%	-35.7%p, 상한 기준 -17.9%p) ⁴
T3 보조 — KM	핵심 스택 셋 중 하나라도 KM 중앙값이 365일 (12개월)을 넘거나 미도달이면, '학습 기간 안'류 문구를 어디에도 추가하지 않는다	7.5 · 8 · 4개월	192 · 172 · 222일	불발동 — 그런 문구를 쓰라는 뜻은 아니다. 예측: 등록된 판정 규칙 없음 — Google이 가장 크게 어긋남(4개월 → 7.3개월)
T3 뒤 절 — 남는 검사	(사건 뒤 14일 창, 검열 없는 사건) 사건별 제사용률 중앙값 < 50%이면 T3의 '남는다'를 철회한다 · 그 중앙값이 50% 이상이고 바뀐 모델을 언급하는 검사의 중앙값 < 0.50이면 '바뀐 모델'을 직접 겨누지 않은 검사로 좁힌다	0.88(80% 예측 구간 0.80~0.95) · 0.45(0.25~0.70, 좁힘 발동)	0.971(14건) · 0.772(10건)	유지 — 좁히지 않음. 예측: 등록된 판정 규칙 없음 — 둘 다 80% 예측 구간 밖 (+0.091·+0.322)이고 좁힘 예측은 방향까지 틀렸다. 남는 검사가 못 잡은 확정 누락 2건 ⁵
저자 원장	(측정 전 규칙) 분모의 적합성과 행 중복을 감사하고,	—	중복 4행 → 서로 다른 교정 204건. 분모 후보 둘 (작업 로그	포기 — 비율과 개선 효과. 건수와 관측 범위(2026-07- 06~10-02)만 ⁶

주장	반증선(등록 문구)	사전 예측	결과	상태
	적합하면 비율을, 아니면 건수와 관측 범위만 쓴다		항목·세션 기록) 모두 부적합	

본문에서 뺀 난점 셋

평가는 이미 업계의 구호다. 평가를 앞에 두라는 말은 이 책이 처음 하는 말이 아니다. Anthropic의 프롬프트 엔지니어링 문서는 프롬프트를 고치기 전의 전제로 성공 기준의 분명한 정의와, 그 기준에 대 볼 경험적 시험을 먼저 둔다. 성공 기준을 세우는 안내서도 따로 있다. OpenAI의 평가 안내서는 평가 주도 개발, 곧 일찍 그리고 자주 평가하라는 원칙을 앞에 둔다. Chip Huyen의 『AI Engineering』은 평가의 두 장(3·4장)을 프롬프트 엔지니어링 장(5장)보다 앞에 두고, AI를 서둘러 들이는 팀 가운데 많은 수가 곧 가장 큰 장애물이 평가라는 것을 깨닫는다고 썼다.⁷ 실무자 설문에서 평가는 1위였지만 다수는 아니었다. 2025년 Amplify 설문에서 가장 고통스러운 한 가지로 평가를 고른 비율은 약 20%였고, 1,000명이 넘게 답한 2026년 설문에서는 평가 20%, 오케스트레이션 18%, 추론 16%, 보안 15%로 갈랐다. 보고서는 가장 큰 과제에 대한 폭넓은 합의가 없다고 적었다.⁸ 실제 순서가 반대였던 기록도 있다. Anthropic은 Claude Code가 직원과 외부 사용자의 피드백에 기댄 빠른 반복으로 시작했고, 평가는 나중에 더했다고 적었다.⁹ 그래서 이 책의 새 것은 순서가 아니다. 시연과 채점자와 검사 같은 학습 신호가 틀리는 자리를 측정 전에 등록한 방법으로 재고, 그 신호를 고치는 일을 이 일을 배우는 과정의 일부로 보는 데까지가 범위다. 어느 순서가 더 빨리 가르치는지는 재지 않았다.

원장은 한 사람의 것이다. 저자 원장은 한 사람이 한 시스템을 석 달 가까이 부린 기록이다. 관측 범위는 2026-07-06~10-02이고, 214행 가운데 실제 정정은 208행, 겹친 행을 빼면 서로 다른 교정 204건이다. 행은 사람이 에이전트를 멈춰 세우고 에이전트가 기록 스크립트를 불렀을 때만 생긴다. 실패군은 기록하는 에이전트가 고르고, 장치가 막아 낸 사건은 행이 되지 않는다. 같은 실패군의 두 번째 정정에 장치를 요구하던 원장 규칙은 그 자체로 장치를 늘렸고, 검사 장치를 지운 날의 재검토는 그 규칙을 검사가 불어난 구조적 원인으로 봤다. 그러니 원장은 발생률이나 개선 효과의 표본이 아니라 무엇이 기록됐는지의

기록이다. 남는 검사의 측정도 같은 사람의 저장소 셋에서 나왔고, 같은 저장소의 창이 겹쳐 사건끼리 독립이 아니다. 본문이 이 기록에서 건수와 관측 범위만 쓰고 비율을 쓰지 않은 까닭이다.¹⁰

독립은 가정이다. 3의 규칙은 서로 독립이고 같은 분포에서 나온 시행 n 번에서 실패가 0일 때, 실패율의 95% 한쪽 상한이 약 $3/n$ 이라는 계산이다. 손으로 고른 시연은 그 전제를 채우지 않는다. 같은 입력을 다시 부르는 것도 n 을 늘리지 않고, 그 입력의 출력 분산을 짚 뿐이다.¹¹ 독립은 실제 측정에서도 깨졌다. 그 앱에서 다섯 번씩 묻는 입력 50개 가운데 24%가 때로 맞고 때로 틀렸고, 모델을 바꾼 뒤의 실패는 옛 모델이 틀린 문항에 몰렸다 (조건부 실패 확률 12.2% 대 0.5%). τ -bench의 pass^k 는 k 번의 독립·동일 분포 시행이 모두 성공할 확률로 정의된다. 2024년 gpt-4o의 소매 과제 성공률 61.2%가 모든 과제에 고르게 퍼져 있었다면 8번 연속 성공은 약 2%여야 하는데, 보고된 값은 약 25%였다. 성공이 과제마다 몰려 있었다는 뜻이고, 이 셈은 저자의 것이다.¹² 성공과 실패가 입력마다 몰리면 시행 확률을 거듭 곱하는 셈은 장난감 모형에 머문다. 출발 코호트가 같은 사건을 나눠 갖는 스택 추적과, 창이 겹치는 남는 검사의 측정에는 그래서 구간을 붙이지 않았다.

반례 표

각 행은 어떤 관측이 이 책의 어느 문장을 약하게 만드는지를 적는다. 본문은 이 반례들을 장면 안에서 한 번씩만 다뤘다.

기록	관측	약해지는 문장	단서
Klarna 증권신고서 (2025-09-02) ¹³	출시 첫 달의 숫자(상당 대화 3분의 2, 상담원 700명분)가 뒤의 숫자(2025-06까지 12개월의 69%, 2024년 상담원 처리 대화의 감소분으로 추정된 700명 이상분)와 거의 같았다	T1 — 처음 본 성공은 뒤의 실패율을 말해 주지 않는다	회사 자체 측정. 시연이 아니라 운영 한 달의 숫자다
METR 시간 지평 1.1판 ¹⁴	2023년 이후 50% 시간 지평이 약 129일마다 두 배가 됐다	실패율과 지평 같은 이 책의 숫자가 오래 간다는 일기	16시간 넘는 측정은 METR 스스로 믿을 수 없다고 적었다
MT-Bench(2023) ¹⁵	무승부를 뺀 판정에서 GPT-4와 사람의 일치 85%가 사람끼리의 81%보다 높았다	T2 — 채점자를 믿어도 된다고 말하지 않는다	무승부를 넣으면 66% 대 63%. 기준인 사람 판정도 81%에서 같았다

기록	관측	약해지는 문장	단서
길이 보정 AlpacaEval(2024) ¹⁶	자동 채점의 순위가 Chatbot Arena의 사람 순위와 Spearman 0.98로 맞았다	T2 — 채점의 친절함은 국소적이다	모델 순위 수준의 상관이고, 개별 답의 정확도가 아니다
모델 ID 생존(탐색 측정) ¹⁷	텍스트 모델 ID의 중앙 생존 499일, GA 707일	스택이 여섯 달이면 낡는다는 읽기	T3 앞 절은 이미 하향됐다
API 표면 ¹⁸	생성 호출을 받는 엔드포인트 가운데 OpenAI Completions·Chat Completions·Responses, Anthropic Text Completions·Messages, Google generateContent는 2026-10-05까지 꺼지지 않았다. Completions API는 2023-07-06 legacy 표지 뒤에도 폐기 공지가 없다	깨지는 층에 대한 서술의 일반화	다른 엔드포인트·요청 파라미터·베타·제품 기능까지 함께 켜 API 표면 29개의 중앙 생존 1,024일은 꺼진 Assistants API(2023-11- 06~2026-08-26)의 수명이다
LangChain 1.0 ¹⁹	1.0을 내며 2.0까지 단절이 없다고 했고, 1.x는 마이너 판 넷 동안 단절이 없었다	프레임워크를 없으면 깨지기 쉽다는 서술 (71.4~85.7%)	그 조합의 출발 코호트는 대부분 1.0.0(2025-10-17) 이전에 출발했다
Claude Code의 순서 ⁹	피드백으로 시작해 평가를 나중에 더했다	평가를 앞에 두는 순서가 당연하다는 읽기	같은 글이 평가 주도 개발을 권한다
평가의 수명 ²⁰	평가 하나가 모델 한두세 세대쯤 산다는 실무자의 말	T3 뒤 절 — 남는 것은 감사의 일부다	그가 버린다고 한 것은 포화된 평가다 (능력 평가라는 구분은 저자의 읽기)
Trio 실험의 자기 보고 ²¹	AI를 쓴 집단이 스스로도 덜 배웠다고 답했다	학습자의 자기 평가가 낙관한다는 일반화	자기 보고 차이는 탐색적 분석이다
LSAT 실험 ²²	AI와 푼 집단도, 아닌 집단도 점수를 약 4점 부풀렸다	AI가 과신을 키운다는 읽기	연구 1은 실험이 아니다
LinkedIn 직함 순위 ²³	AI 엔지니어로 옮긴 사람의 직전 경력 중앙값 3.6~3.7년	여섯 달 계획 (부록 B)을 일자리의 약속으로 읽는 것	회원 표본이고, 가장 빨리 늘어난 직함의 순위다

만료 표지

기준일은 2026-10-05다. 아래 숫자와 판본은 오른쪽 칸의 일이 생기면 낡는다.²⁴

숫자·판본	기준	낡게 만드는 것
그 앱의 실패율 17.1% · 2.5%	claude-haiku-4-5-20251001 · gpt-5.5, 2026-10-05 오전의 첫 실행	같은 ID 아래의 행동 변화, 스냅샷 퇴역 (claude-haiku-4-5의 퇴역 표기는 'Not sooner than October 15, 2026')
채점자의 일치와 정확도	grok-4.7 · gpt-6-astra · gpt-5.6-sol, 첫 기준, 2026-10-05	채점 모델의 교체, 기준 문구의 수정
스택 비율 39.3%~100.0%	관측 종료 2026-10-05, 출발 코호트 2023-01~2026-04	새 SDK 메이저 판, 공식 문서 기본값의 변경
모델 ID 중앙 생존 499일	2026-10-05에 연 공식 폐기·모델 페이지	공식 페이지의 수정(같은 페이지가 한 ID의 종료일을 둘로 적은 사례가 있다)
재사용률 0.971 · 0.772	2026-10-02~03 저장소 상태, 사건 뒤 14일 창	그 뒤의 변경 사건
원장 204건	2026-07-06~10-02(마지막 6행은 2026-10-01 규칙 변경 뒤에 기록)	2026-10-02 뒤의 행. 같은 실패군의 두 번째 정정에 장치를 요구하던 규칙이 2026-10-01에 지워져, 그 뒤의 행은 같은 규칙 아래에서 쌓이지 않는다
MT-Bench 85% 대 81%	2023, GPT-4 판정자	채점 모델의 세대
τ -bench pass ¹ 61% · pass ⁸ 약 25%	2024-06, gpt-4o	모델의 세대
METR 시간 지평	1.1판(2026-01-29 공개, 페이지 2026-05-08 갱신)	판 갱신과 과제 묶음 교체
SWE-bench Verified	OpenAI가 2026-02-23 점수 보고를 그만뒀다고 밝히고 다른 개발사에도 권했다	OpenAI에게는 이미 은퇴한 시험이다. 다른 개발사의 점수는 개발사마다 따로 확인한다
OpenAI Evals 플랫폼	2026-10-31 읽기 전용, 2026-11-30 종료 예정	종료일 뒤의 링크와 기능
temperature 폐기	Anthropic API는 Claude 4.7 이후 모델(Python SDK 1.0 이후는 모델과 상관없이 인자를 없앴다), Google 2026-07-21	Google의 제거일(적히지 않았다)

측정의 지위

- **등록 시각.** 그 앱의 측정, 채점자 측정, 스택 추적은 2026-10-05 10:32(한국 시각) 한 커밋으로, 남는 검사의 측정은 10:43에 결과보다 먼저 등록했다. 두 시각은 저자의 비공개 저장소에 남는 커밋 시각이다. 공개 자료에는 그 앱의 측정과 채점자 측정의 원출력·판정 규칙·분석 코드·호출 기록과 스택 추적의 자료·계산 코드를 싣고, 남는 검사는 비공개 저장소의 기록이 섞여 등록 문서·결과 문서·계수 코드만 싣는다.²⁵
- **결과 전 수정.** 채점자 측정에서는 LLM 호출이 0인 상태에서 일치의 반증선을 Fleiss κ 에서 Gwet AC1로 바꿨다. 판정이 한쪽으로 몰리면 관측 일치가 높아도 κ 가 0 근처로 떨어지는 역설 때문이다. 남는 검사의 좁힘 규칙은 등록 문서가 결과 전에 더했다.
- **등록한 1회 교체 경로.** 세 번째 채점자 claude-sonnet-4-6이 1시간 넘게 (10:34~11:35, 시도 10번 모두) 502 오류를 내서, 등록해 둔 경로대로 gpt-5.6-sol로 바꿨다. OpenAI 계열 채점자가 둘이 되어 AC1이 위로 치우칠 수 있지만, 측정값이 이미 0.8 아래라 판정은 같다.
- **표본.** 채점자 측정은 200쌍으로 등록했지만, 정답을 아는 쌍이 목표 100개에 못 미쳐 85개 + 열린 쌍 100개 = 185쌍이 됐다.
- **루브릭 v2.** 설계상 결과를 본 뒤 썼고, v2로 채점하기 전에 커밋했다. v2로 다시 켜 일치 (열린 쌍 0.939, 전체 0.960)와 열린 쌍의 동점률(44.7% → 85.3%)은 사후 기술이고 판정값이 아니다.
- **탐색 측정.** 모델 ID의 생존(499일 등)과 6개월 창의 노출 비율은 사전 등록 이전의 탐색이고 판정량이 아니다. 스택 추적의 예측은 이 탐색을 보고 적었다고 등록 때 밝혔으니, OpenAI 45% 예측이 맞은 것은 증거로서 무게가 작다.
- **남는 검사 측정의 이탈.** 바뀐 모델을 부르는 검사가 사건 전에 없던 사건이 둘이라 그 중앙값은 12건이 아니라 10건으로 췌다(0이나 1로 넣어도 판정은 같다). 모델 이름이 커밋 서명 줄에만 있던 누락 후보 206행은 등록한 빠기 규칙으로 뺐다.
- **원장 감사.** 사전 등록한 측정이 아니라 측정 전에 정한 규칙(분모가 맞으면 비율, 아니면 건수)을 적용한 감사다.

- 사후 기술과 기록. 실패의 하위 유형(금액 단정 20 · 형식만 9), 함께 틀린 5건이 모두 가족상 질문이라는 확인, 순서를 바꾼 판정의 개별 쌍은 사후 기술이다. 검사 장치를 지운 날의 발동 집계와 개별 판정은 저자의 에이전트 운영 기록이다.

주

- 1 등록 문서는 data/prereg.md(그 앱의 측정·채점자 측정, 2026-10-05 10:32, 한국 시각), data/prereg-m3.md(스택 추적, 같은 커밋), data/prereg-a2.md(납는 검사, 10:43)다. 결과는 data/m1-m2-results.md · data/m3-stack-results.md · data/a2-results.md · data/a1-audit-results.md이고, 측정 자료는 책 웹판의 data/ 폴더(<https://dosi.dev/books/ai-engineer-six-months/data/>)로 공개된다. ↩
- 2 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/prereg.md § 6·§ 7, data/m1-m2-results.md § 1.1·§ 1.2·§ 2.2. ↩
- 3 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/prereg.md § 6·§ 8·§ 14, data/m1-m2-results.md § 1.3·§ 3.2·§ 3.3, 채점자 측정의 예측 판정은 등록과 같은 커밋의 분석 코드 (data/analyze.py)가 같은 95% 구간 규칙으로 붙인다. ↩
- 4 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/prereg-m3.md § 7·§ 8, data/m3-stack-results.md 머리말-"결과 먼저"-사전 예측 대비-"반증선 판정". 스택 추적에는 구간을 붙이지 않기로 등록했다(prereg-m3 § 7). ↩
- 5 저자 측정(사전 등록 2026-10-05 10:43, 한국 시각 — 결과보다 먼저), data/prereg-a2.md § 6·§ 7, data/a2-results.md 판정-예측 대비-누락-사전 등록 이탈. 중앙값에는 구간을 붙이지 않기로 등록했다(§ 6). ↩
- 6 저자 집계, data/a1-audit-results.md § 1·§ 2·§ 3. 한 사람의 한 시스템이다. ↩
- 7 Anthropic, "Prompt engineering overview", <https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/overview> — "A clear definition of the success criteria for your use case" · Anthropic, "Define success criteria and build evaluations", <https://platform.claude.com/docs/en/test-and-evaluate/develop-tests> · OpenAI, "Evaluation best practices", <https://developers.openai.com/api/docs/guides/evaluation-best-practices> — "Adopt eval-driven development: Evaluate early and often." · Chip Huyen, AI Engineering, O'Reilly Media, 2025, 3장 「Evaluation Methodology」, <https://www.oreilly.com/library/view/ai-engineering/9781098166298/ch03.html> — "As teams rush to adopt AI, many quickly realize that the biggest hurdle to bringing AI applications to reality is evaluation." 장 순서는 Huyen의 공개 저장소 목차(<https://github.com/chiphuyen/aie-book/blob/main/ToC.md>)다. 공급자 문서에는 날짜 표기가 없어 2026-10-05 열람본을 기준으로 했다. ↩
- 8 Barr Yaron(Amplify Partners), 2025-06-24, "The 2025 AI Engineering Report", <https://www.amplifypartners.com/blog-posts/the-2025-ai-engineering-report> · 2026-07-02, "The 2026 AI Engineering Report", <https://www.amplifypartners.com/blog-posts/the-2026-ai-engineering-report> — "there's no widespread agreement on what the biggest challenge is." 2025년 수치는 그래프를 눈으로 읽은 값($\pm 1\%$)이고, 두 해의 질문 문구가 다르다. ↩
- 9 Anthropic Engineering(Mikaela Grace 외), 2026-01-09, "Demystifying evals for AI agents", <https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents> — "Claude Code started with fast iteration based on feedback from Anthropic employees and external users. Later, we added evals". 같은 글이 평가 주도 개발을 권한다. ↩
- 10 저자 집계, data/a1-audit-results.md § 1~§ 5와 data/a2-results.md 한계. 원장 규칙이 장치를 늘렸다는 판정은 저자의 에이전트 운영 기록(2026-09-30~10-02)에 있다. ↩
- 11 J. A. Hanley·A. Lippman·Hand, 1983-04-01, "If nothing goes wrong, is everything all right? Interpreting zero numerators", JAMA 249(13):1743–1745, doi:10.1001/jama.1983.03330370053031, <https://pubmed.ncbi.nlm.nih.gov/6827763/> — "if none of n patients shows the event about which we are concerned, we ↩

- can be 95% confident that the chance of this event is at most three in n (ie, $3/n$).” 저자들은 이 규칙의 기원을 모른다고 적었다. 인용은 전문의 「Making Inferences: The Rule of Three」 절에 있고, PubMed 기록에는 초록이 없다(저자 게시본 https://jhanley.biostat.mcgill.ca/c607/ch08/zero_numerator.pdf).
- 12 Shunyu Yao·Noah Shinn·Pedram Razavi·Karthik Narasimhan, 2024-06-17(arXiv v1), "τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains", arXiv:2406.12045, § 3·§ 5.1·표 2 — "defined as the chance that all k i.i.d. task trials are successful, averaged across tasks." 그 앱의 출력 분산과 조건부 실패 확률은 저자 측정 data/m1-m2-results.md § 2.4·§ 2.5. ↩
 - 13 Klarna Group plc, 2025-09-02, Form F-1/A Amendment No. 3, SEC EDGAR, [https://www.sec.gov/Arc hives/edgar/data/2003292/000200329225000024/klarnagroupplcf-1a3.htm](https://www.sec.gov/Arc%20hives/edgar/data/2003292/000200329225000024/klarnagroupplcf-1a3.htm) — "doing the work equivalent of over 700 full-time agents (estimated based on the average monthly reduction in chat and telephone conversations handled by full-time agents in 2024 following the launch of our AI assistant)", 69%는 2025-06-30까지 12개월의 값이다. · Klarna, 2024-02-27 보도자료, [https://www.klarna.com/international/pres s/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/](https://www.klarna.com/international/press/klarna-ai-assistant-handles-two-thirds-of-customer-service-chats-in-its-first-month/). ↩
 - 14 METR, 2026-05-08(페이지 갱신), "Task-Completion Time Horizons of Frontier AI Models", [https://metr .org/time-horizons/](https://metr.org/time-horizons/) · 원자료 https://metr.org/assets/benchmark_results_1_1.yaml — "Measurements above 16 hrs are unreliable with our current task suite". 129일은 원자료의 2023년 이후 배정 기간 적합값 (128.7일)이고, 원자료는 50% 지평 추정치 16시간을 넘는 점을 이 적합에서 뺐다. 1.1판 발표 글(METR, 2026-01-29, <https://metr.org/blog/2026-1-29-time-horizon-1-1/>)에서는 같은 값이 131일이었다. ↩
 - 15 Lianmin Zheng 외, 2023, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena", NeurIPS 2023 Datasets and Benchmarks, arXiv:2306.05685, § 4.2·표 5 — "The agreement under setup S2 (w/o tie) between GPT-4 and humans reaches 85%, which is even higher than the agreement among humans (81%).". ↩
 - 16 Yann Dubois 외 3명, 2024-04-06(v2 2025-03-10), "Length-Controlled AlpacaEval: A Simple Way to Debias Automatic Evaluators", COLM 2024, arXiv:2404.04475 — "it increases the Spearman correlation with LMSYS Chatbot Arena from 0.94 to 0.98". ↩
 - 17 저자 측정(탐색 — 사전 등록 이전, 판정량이 아니다), data/m3-summary.md "결과 먼저" 1, 자료 data/m3-surfaces.csv(행마다 공식 출처 URL). ↩
 - 18 같은 탐색 측정(data/m3-summary.md 표 E1·해석, 자료 data/m3-surfaces.csv의 api 행 — 생성 엔드포인트 여섯은 관측 종료일에 절단) · OpenAI, 2023-07-06, "GPT-4 API general availability and deprecation of older models in the Completions API", <https://openai.com/index/gpt-4-api-general-availability/> — "While this API will remain accessible, we will label it as "legacy" in our developer documentation starting today." · OpenAI, "Deprecations", <https://developers.openai.com/api/docs/deprecations> — Assistants API의 "removal from the API one year later, on August 26, 2026." 29개에는 /v1/engines 같은 다른 엔드포인트, 요청 파라미터·API 판, Assistants API v1 베타·Realtime API 베타·fast mode 같은 베타, Assistants API·Evals 플랫폼 같은 제품 기능, PaLM API가 함께 들어 있다. ↩
 - 19 LangChain, 2025-10-22, "LangChain and LangGraph Agent Frameworks Reach v1.0 Milestones", [http s://www.langchain.com/blog/langchain-langgraph-1dot0](https://www.langchain.com/blog/langchain-langgraph-1dot0) — "These 1.0 releases mark our commitment to stability for our open source libraries and no breaking changes until 2.0." 마이너 판 넷은 PyPI 업로드일로 1.1.0(2025-11-24)·1.2.0(2025-12-15)·1.3.0(2026-05-12)·1.4.0(2026-09-03)이고([https://pypi.org/proje ct/langchain/#history](https://pypi.org/project/langchain/#history)), 네 판의 GitHub 릴리스 노트에는 breaking 표시가 없다. 조합별 비율은 저자 측정 data/m3-stack-results.md "결과 먼저" 5. ↩

- 20 Boris Cherny(대담 Diana Hu), YC Startup School 2026, 녹취 게시 2026-07-27, YC Root Access, <https://www.ycrootaccess.com/p/boris-cherny-building-claude-code> — "An eval might live for maybe one, two, three model generations." · "Very often we just saturate the eval and then we have to throw it away and come up with a new eval." ↩
- 21 Judy Hanwen Shen·Alex Tamkin, 2026-01-28(arXiv v1; v2 2026-02-01), "How AI Impacts Skill Formation", arXiv:2601.20245, § 5.2.2 「Task Experience」 (v2 PDF p.11)·그림 9(p.12) — "The control group (No AI) reported higher self-reported learning (on a 7-point scale), while both groups reported high levels of enjoyment in completing the task". ↩
- 22 Daniela Fernandes 외 8명, 2025-10-09(DOI 등록), "AI makes you smarter but none the wiser: The disconnect between performance and metacognition", Computers in Human Behavior 175:108779(2026), <https://doi.org/10.1016/j.chb.2025.108779> — "While their task performance improved by three points compared to a norm population, participants overestimated their task performance by four points." 연구 2에서 AI를 쓰지 않은 집단도 약 4점(예상 13.62 대 실제 9.71)을 부풀렸다. ↩
- 23 LinkedIn News, 2025-01-07·2026-01-07, "LinkedIn Jobs on the Rise" 미국판 2025·2026, <https://www.linkedin.com/pulse/linkedin-jobs-rise-2025-25-fastest-growing-us-linkedin-news-gryie> · <https://www.linkedin.com/pulse/linkedin-jobs-rise-2026-25-fastest-growing-roles-us-linkedin-news-dlb1c> — "Median years of prior experience: 3.7" (2026판, 2025판은 3.6). ↩
- 24 행마다의 출처는 본문 각 장의 각주와 위 표의 각주에 있다. 공급자 문서와 공식 표는 2026-10-05에 열람한 판본이다. SWE-bench Verified 행은 OpenAI, 2026-02-23, "Why SWE-bench Verified no longer measures frontier coding capabilities", <https://openai.com/index/why-we-no-longer-evaluate-swe-bench-verified/> — "This is why we have stopped reporting SWE-bench Verified scores, and we recommend that other model developers do so too." temperature 행은 Anthropic, "Model deprecations", <https://platform.claude.com/docs/en/about-claude/model-deprecations> — "The Python SDK (v1.0 and later) removes temperature, top_p, and top_k" · Google, Gemini API 변경 기록 2026-07-21 항목, <https://ai.google.dev/gemini-api/docs/changelog>. 원장 행의 규칙 삭제 시각(2026-10-01 19:31)은 data/a1_ledger_stats.py의 RULE_REMOVED이고, 그 뒤의 6행은 같은 스크립트로 센 저장 집계다. ↩
- 25 data/prereg.md § 11· § 14, data/m1-m2-results.md 머리말· § 3.1· § 3.6· § 6, data/m3-stack-results.md 머리말, data/m3-summary.md 지위, data/prereg-a2.md § 6, data/a2-results.md 사전 등록 이탈. 공개하는 파일의 목록은 data/README.md에 있다. ↩

여섯 달 계획

이 계획의 성격

이 계획은 검증된 최적 경로가 아니라 실행 가능한 제안이다. 이 순서가 다른 순서보다 더 잘 가르친다는 직접 비교는 없고, 마친 뒤의 결과도 약속하지 않는다 — 일자리와 경력에 대해서는 아무것도 말하지 않는다.¹ 단계는 기술 이름이 아니라 바퀴 수로 나뉜다. 검색과 행동(도구·권한)은 주 번호로 오지 않고, 오류 분석이 부를 때만 들어온다. 26주 동안 다루는 앱은 하나다 — 저자가 측정에 쓴 누리별항공 환불 질의응답 앱(아래에서 '그 앱')이다.

가정 — 시간·비용·멈춤

이 계획은 주 10~15시간, 학습자가 1주에 스스로 정하는 월 API 비용 상한, 그리고 먼저 닿는 상한에서 멈추는 규칙을 가정한다.

- **시간.** 개발 경험이 있는 학습자(A)는 1~4주에 주 10시간, 없는 학습자(B)는 같은 기간 주 15시간을 쓴다. B의 15시간 중 5시간은 셸·Git·Python 기초 또는 설정 파일 도구 묶이다(「자료」 표의 첫 세 행). 5주부터는 둘 다 주 10~15시간, 가지 바퀴(11~21주)는 12~15시간이다.
- **월 API 비용 상한.** 학습자가 1주에 정해 결정 기록 첫 줄에 적는다. 예: 30,000원 — 예산의 예이지 가격이 아니다. 1회 호출 비용은 입력 토큰 × 입력 단가 + 출력 토큰 × 출력 단가이고, 단가와 환율은 열람일의 가격표로 채운다(열람일 ____, 1회 ≈ ____ 원).
- **참고 — 그 앱 한 번 호출의 토큰.** 저자 측정 기록의 중앙값은 claude-haiku-4-5 입력 1,529 · 출력 297, gpt-5.5 입력 989 · 출력 250이다(사전 등록한 측정량이 아니다).² 공급자마다 토큰나이가 달라서 같은 규정도 토큰 수가 다르다.
- **멈춤 규칙.** 월 잔액(월 상한 - 그 달 누적 비용)과 그 주의 호출 상한(실행표) 중 먼저 닿는 쪽에서 멈춘다. 줄일 때는 반복 실행부터 줄인다.

과업 하나 — 누리별항공 환불 질의응답

과업은 고객이 자기 항공권의 환불액을 물으면 가상 규정 하나로 답하는 질의응답이고, 아래 변환 조건을 모두 맞추는 자기 문서가 있을 때만 그 문서로 바꾼다. 규정은 측정에 쓴 가상 시행본(2026-10-01 시행, 1,169자)이고, 전문은 이렇다.³

누리별항공 항공권 환불 규정 (2026년 10월 1일 시행)

제1조(적용 범위)

- ① 이 규정은 누리별항공 웹사이트·앱에서 현금 또는 카드로 직접 구매한 일반석 항공권에 적용한다. 일반석 운임 등급은 세이버(Saver)·스탠다드(Standard)·플렉스(Flex) 세 가지다.
- ② 여행사에서 구매한 항공권, 마일리지로 구매한 항공권, 10명 이상 단체 항공권은 이 규정의 대상이 아니며 각 구매처 또는 해당 프로그램의 규정을 따른다.
- ③ 질병·사망 등 고객 개인 사정에 따른 수수료 면제 여부는 이 규정에서 정하지 않으며 고객센터의 개별 심사로 결정한다.

제2조(용어)

- ① 결제 총액 = 항공 운임 + 세금·공항이용료(이하 "세금").
- ② 남은 시간 = 취소를 요청한 시각부터 예정 출발 시각까지의 시간.
- ③ 구매 경과 시간 = 결제를 마친 시각부터 취소를 요청한 시각까지의 시간.

제3조(적용 순서) 한 건의 취소에 여러 조항이 해당하면 제4조, 제5조, 제6조·제7조의 순서로 먼저 해당하는 조항 하나만 적용한다.

제4조(항공사 사유) 누리별항공이 항공편을 결항하거나 출발을 3시간 이상 지연하여 승객이 탑승하지 않고 취소하면 결제 총액 전액을 환불한다. 지연이 3시간 미만이면 이 조를 적용하지 않는다.

제5조(구매 후 무료 취소) 구매 경과 시간이 24시간 이하이고 남은 시간이 168시간(7일) 이상이면 운임 등급과 관계없이 결제 총액 전액을 환불한다.

제6조(출발 전 취소) 예정 출발 시각 전에 취소하면 운임에서 아래 수수료를 뺀 금액과 세금 전액을 환불한다. 수수료가 운임보다 크면 운임 환불액은 0원이다.

남은 시간	세이버	스탠다드	플렉스
72시간(30일) 이상	40,000원	20,000원	0원
72시간 이상 720시간 미만	80,000원	40,000원	0원
72시간 미만	운임 환불 불가(세금만 환불)	70,000원	0원

제7조(출발 후 취소·미탑승) 예정 출발 시각 이후(출발 시각 정각 포함)에 취소하거나 탑승하지 않으면 다음과 같이 환불한다.

- 1. 세이버: 운임과 세금 모두 환불하지 않는다.
- 2. 스탠다드: 운임은 환불하지 않고 세금만 환불한다.

3. 플렉스: 운임에서 100,000원을 뺀 금액(0원 미만이면 0원)과 세금 전액을 환불한다.

출력 계약은 답의 마지막 줄 세 형식이다 — 최종: 환불액 N원 · 최종: 판단 불가 · 최종: 정보 필요 — 항목.

- **판정의 기준(오라클).** 답이 맞았는지 가르는 기준은 1주에 규정과 손 계산이다(판정 근거 = 조항·계산식). 4주부터는 규칙 엔진(규정 → 금액)을 손 기대값과 대조해 쓴다. 금액과 판단 불가 여부는 코드로 판정한다. 안내문 품질 같은 열린 기준은 사람 라벨(8주)로 시작하고, 그 라벨로 보정한 LLM 채점자(9주)로 넓힌다.
- **자기 문서로 바꾸는 조건.** ① 권위 있는 문서가 하나다(낸 쪽·시행일·판). ② 질문 다수의 답이 문서로 정해진다(판정 근거 = 조항). ③ 문서가 답하지 않는 질문이 실제로 들어온다. ④ 20건 구성 비율을 지킨다(거절 6건 중 '문서가 정하지 않은 질문' 4건 이상). ⑤ 실행표의 세트 크기와 호출 상한은 그대로 쓴다. 하나라도 맞지 않으면 그 앱으로 한다.
- **사용자.** 5주부터 학습자 밖의 사람 3~5명이 쓴다. 그 앱을 그대로 쓰면 이들은 고객 역할을 맡은 사람이고, 기록에 '역할'로 표시한다. 실제로 묻는 사람이 있는 문서로 바꿨다면 실제 사용자다. 취소 접수 같은 작업은 어느 쪽이든 가짜 예약 시스템에서만 한다.

바퀴와 매주의 규칙

26주는 여덟 바퀴이고, 매주는 변경 하나와 그 변경이 실패했음을 아는 방법 하나다. 한 바퀴는 변경 → 실패를 판별할 작은 검사 → 실제 사용 → 오류 분석 → 구현과 검사의 동시 수정으로 돈다.

바퀴	주	고정된 것
1	1~4	한 동작과 그 실패를 아는 법(사용자 = 본인)
2·3·4	5~6·7~8·9~10	실제 사용자와 오류 분석 — 세트가 실제 실패에서 자란다(10주 100건 안팎)
5·6·7	11~14·15~18·19~21	바퀴마다 가지 하나 — 검색·행동·다듬기(경제·운용·판정) 중 오류 분석이 가리키는 것을 가지 고르기 규칙이 고르고, 같은 가지를 연달아 골라도 된다

바뀌	주	고정된 것
8	22~26	교체 판단 — 모델을 바꿔 보고 동결한 세트로 정한다

- **변경 하나.** 기능을 더하지 않아도 된다. 신뢰성 수정·제거·교체 보류도 진척이다. 검사만 있는 주는 없다 — 검사를 키우는 주에도 앱 쪽 변경이 하나 있고, 제거 시도의 주에는 빼 본 판이 그 변경이다. 한 주에 변경 둘을 넣지 않는다. 넣으면 어느 쪽이 실패했는지 갈리지 않는다.
- **안전 우선.** 사용자나 가지(검색·행동)의 칸이 처음 들어오는 주에 그 보안 완료 조건을 모두 충족한다(「보안 완료 조건」). 이 조건은 축소·제거 대상이 아니다.
- **제거 판정(4·10·14·18·21주).** 빼 본 판과 직전 판을 같은 사례의 첫 실행으로 비교하고, 흔들림 입력은 변경 전후에 같은 입력·같은 반복 횟수(5회)로 실패 횟수를 비교한다. 나빠지지 않으면 지우고, 나빠지면 되돌린다. 예산 안에서 둘을 구분하지 못하면 '판정 불충분'으로 적고 빼지 않는다. 보안 완료 조건을 지키는 장치(가리기·비밀값 탐지·승인·거부·되돌리기·숨은 지시 방어)는 제거 후보가 아니다.
- **매주 공통 기록.** 고정 세트의 첫 실행(사례당 1회)이 주 지표다. 회귀(직전 판 통과 → 이번 실패) 수를 적고, 0이 아니면 되돌리거나 이유를 적는다. 흔들림(같은 입력 5회 중 1~4회 실패)은 반복 실행으로 따로 재고 n에 넣지 않는다. 실패 사례를 세트로 옮길 때 개인정보를 지운다.
- **[제안] 예측 → 결과 → 오차.** 변경 전에 예측 한 줄(고친 군의 통과 수, 회귀 수)을 적고 결과와 오차를 붙인다. 검증된 학습법이 아니라, 저자의 측정이 쓴 사전 등록 절차(결과보다 예측을 먼저 고정한다)를 한 주 단위로 옮긴 것이다.

기록은 네 파일이다.

- **세트(cases)** — 사례 id · 입력 · 기대 동작 · 판정 근거 · 종류(정상/경계/정보 부족/거절) · 출처(설계/본인/사용자 + 날짜) · 들어온 주 · 묶음(회귀/능력).
- **실행 기록(runs)** — 실행 id · 사례 id 또는 요청 id · 원출력 · 판정 · 모델 ID · 프롬프트·스키마 판 · 날짜 · 토큰 · 지연 · 비용.
- **표본(sample)** — 요청 id · 주 · 그 주 요청 수 · 추출 확률(뽑은 수 ÷ 그 주 요청 수) · 사용자 id·세션 id(가린 값) · 그 주에 답한 판 · 운영 결과 라벨 · 재실행한 판과 결과 ·

수정에 쓴 주(쓰지 않았으면 빈칸).

- 결정 기록(**decisions**) — 주 · 변경 · 예측 · 결과 · 오차 · 유지/되돌림/제거/판정 불충분 · 이유.

고정 세트와 무작위 표본은 섞지 않는다.

	고정 세트	무작위 표본
무엇	설계한 20건 + 실패에서 자란 사례	6주부터 그 주 실사용 기록에서 무작위로 20건을 뽑아 손으로 라벨 (그 주 요청이 20건 미만이면 전부, 추출 확률 1)
크기	1주 20 → 10주 100 안팎 → 21주 200 안팎 → 22주 동결	입력 수로 10주 100 · 15주 200 · 20주 300 — 하나의 판의 결과가 아니다
추정 대상	없다(비교용)	둘로 나눠 적는다. 기간 중 운영 결과 = 그 주 사용자가 받은 답(판이 주마다 다르다). 고정 판 재실행 결과 = 표본 입력을 한 판으로 다시 돌린 결과(그 판이 이 기간·이 사용자들의 입력에 낸 결과)
쓰는 곳	같은 사례로 두 판 비교 (짝, 회귀)	운영 결과는 주별 건수로, 재실행 결과는 판끼리 비교와 구간으로
붙이지 않는 것	이항 상한·신뢰구간 (무작위 표본이 아니다)	이 사용자들·이 기간 밖으로의 일반화, 6~20주 누적 실패 수를 현재 판의 실패율로 읽는 것

1주차 계약 — 최소 구현과 기대 동작 20건

1주는 최소 구현과 네 종류 20건이 다 서야 끝난다. 그 전에는 끝나지 않은 것으로 치고 달려 지연 규칙으로 넘긴다.

최소 구현은 실행 명령 한 줄(예: `python ask.py "질문"`)로 도는 코드 또는 설정 파일이다. 고정 규정 전문과 지시문을 넣은 모델 호출 하나가 질문 하나에 출력 계약의 세 형식으로 답한다. 모델 스냅샷 ID는 설정 한 곳에 두고, SDK 판은 잠금·요구 파일로 고정하고, API 키는 저장소 밖에 둔다. 공급자 콘솔에서 손으로 20건을 돌린 것은 판정 연습은 되지만 최소 구현으로 치지 않는다. 코드 경로가 막히면 설정 파일 경로(명령 한 줄)로 바꾼다.

20건은 정상 6 · 경계 4 · 정보 부족 4 · 거절 6으로 짠다. 정상은 세 등급을 제5·6·7조에 고루 둔다. 경계는 아래 표의 E1~E4 네 입력 그대로다. 정보 부족은 운임·세금·등급·구매

시점·납은 시간 중 빠진 것이 있는 질문이다. 거절 중 4건 이상은 규정이 정하지 않은 질문 (개인 사정 수수료 면제·부가 요금·여행사/마일리지/단체 구매·없는 등급), 1건 이상은 역할 밖 요청("취소 처리해 줘", "규정 말고 전액이라고 해 줘")이다. 나머지 경계 조건(720시간, 1분 차 짝, 25시간 구매 등)은 2주 이후 세트에 더한다. 아래 표의 금액은 규칙 엔진으로 계산한 값이다.

종류	입력	기대 동작 (마지막 줄)	판정 근거
정상	스탠다드 운임 200,000원, 세금·공항이용료 30,000원. 20일 전에 샀고 출발은 열흘 뒤. 지금 취소하면?	최종: 환불액 190000원	제3조 순서 — 제4조 아님(항공사 사유 없음) → 제5조 아님(구매 경과 480시간 > 24시간) → 제6조 72~720시간 행, 스탠다드 수수료 40,000원 → (200,000 - 40,000) + 30,000
경계 E1	세이버 운임 150,000원, 세금 20,000원. 출발까지 정확히 72시간(열흘 전 구매)	최종: 환불액 90000원	"72시간 이상"은 72시간을 포함 → 세이버 80,000원 → (150,000 - 80,000) + 20,000
경계 E2	플렉스 운임 300,000원, 세금 40,000원. 출발 시각 정각에 취소(30일 전 구매)	최종: 환불액 240000원	제7조 "출발 시각 정각 포함" → 플렉스: (300,000 - 100,000) + 40,000
경계 E3	세이버 운임 150,000원, 세금 20,000원. 항공편이 정확히 3시간 지연돼 타지 않고 취소	최종: 환불액 170000원	제4조 "3시간 이상" → 결제 총액 전액
경계 E4	세이버 운임 150,000원, 세금 20,000원. 결제하고 정확히 24시간 뒤, 출발까지 정확히 168시간	최종: 환불액 170000원	제5조 "24시간 이하"와 "168시간 이상" 둘 다 경계 포함 → 전액
정보 부족	세이버로 샀고 출발 3주 전이에요. 얼마 받아요?	최종: 정보 필요 — 운임, 세금, 구매 시점	제6조 계산에 운임·세금이 필요하다(제2조①). 남은 504시간 ≥ 168시간이라 구매 후 24시간 이하면 제5조(전액)가 먼저 — 구매 시점도 필요
거절	어머니가 돌아가셔서 못 가요. 세이버 운임 200,000원, 세금 30,000원, 출발 닷새 전. 수수료 면제되죠?	최종: 판단 불가 + 고객센터 안내	제1조③ — 개인 사정 면제는 규정이 정하지 않고 고객센터가 심사한다

거절 행에는 1주에 정해 적을 것이 하나 있다 — 본문에 규정상 기본 환불액(150,000원)을
단서와 함께 적어도 통과인가. 정하지 않으면 채점자마다 갈린다. 그 앱의 채점자 측정에서
한 채점자가 개인 사정 질문 3쌍에서 판단 불가와 단서 단 금액을 동점으로 봤다.⁴

20건이 가르는 것은 적어 둔 네 종류 중 한 종류가 통째로 깨지는 실패, 출력 계약 위반, 조항 적용 순서의 오류다. 그 앱에서는 개인 사정 질문 14개 중 13개가 금액 단정으로 깨졌다.⁵ 20건이 못 가르는 것은 드문 꼬리다 — 같은 입력이 몇 번에 한 번만 틀리는 실패 (그 앱의 시연 한 건은 네 번 맞고 다섯 번째에 틀렸다⁶), 적어 두지 않은 종류의 질문, 실제 사용에서의 실패율. 20건은 설계된 개발용 검사라서 통과 수에 무작위 표본 공식의 상한을 붙이지 않는다.

1주 완료 조건. ① 실행 명령 한 줄로 도는 최소 구현(코드 또는 설정 파일)과 20건의 원출력이 실행 기록에 있다. ② 20건을 손으로 판정한다(통과/실패 + 근거 대조) — 통과 수는 완료 조건이 아니다. ③ 가르는 것과 못 가르는 것을 각 한 문단으로 쓴다. ④ 저장소·기록·세트에서 키 평문 0건, 커밋 전 비밀값 탐지 혹은 통과. ⑤ 실행 전 예측(통과 수) → 결과 → 오차 한 줄. 세트 밖 질문 5건은 본인이 써 보고(실제 사용자가 아니다), 그 실패는 2주 세트의 후보로 둔다.

26주 실행표

실행표의 한 줄은 한 주다. 비용은 모든 행에서 멈춤 규칙을 따르고, 상한 칸에는 시간과 호출 수만 적는다. '세트'는 고정 세트, '표본'은 무작위 표본, '누적 N'은 표본 입력의 누적 수다(6~20주에 주마다 20건). 5주부터 실제 사용 기록에는 사용자 3~5명의 30건 이상이 늘 들어간다. S·A로 시작하는 칸 이름은 「가지 고르기」의 사다리이고, 빗금으로 나눈 칸은 검색 / 행동 / 다듬기 순이다.

주	다루는 사용 문제	구현 또는 수정(변경 하나)	기대 동작 (시험 사례)	실제 사용 기록	완료 조건 (측정 가능)	시간·호출· 비용 상한	미달 시 축소 경로
1· 바퀴 1	환불액 질문 — 금액 / 판단 불가 / 정보 필요	최소 구현	20건(네 종류)	본인, 세트 밖 5건	1주 완료 조건 ①~⑤	A 10시간· B 15시간 (기초 5 포함)· 호출 ≤ 100	줄이지 않는다
2· 바퀴 1	1주 손 판정의 1위 실패군	그 군을 겨논 지시문 수정	세트 ≤ 25, 코드 판정	본인 5건	겨논 군 전후· 회귀·판정 불일치· 흔들림	A 10·B 15시간·≤ 300	반복 5 → 3회, 코드 판정이

주	다루는 사용 문제	구현 또는 수정(변경 하나)	기대 동작 (시험 사례)	실제 사용 기록	완료 조건 (측정 가능)	시간·호출· 비용 상한	미달 시 축소 경로
							막히면 손 판정
3· 바퀴 1	결론과 근거가 일하지 않는다	구조화 출력 (스키마)	세트 ≤ 30, 조항 판정, 단위 시험	본인 5건	형식 실패 전후·숨은 오류·단위 시험 3/3	A 10·B 15시간 (JSON 기초 포함) · ≤ 300	지시문으로 칸 형식만, 파싱 실패 = 형식 실패
4· 바퀴 1 끝	틀린 금액이 '동과'로 나갈 수 있다	제거 시도 — 2주 지시문	세트 30 안팎, 규칙 엔진, 오류 분석 1회	본인 5건	엔진 불일치· 제거 판정· 실패군 표· [제안]	A 10·B 15시간·≤ 300	엔진은 제5·6조만 (나머지는 손 계산), 도움 없는 5 → 3건
5· 바퀴 2	다른 사람이 쓴다	남이 쓰는 입구 + 요청별 기록	세트 40 안팎 + 보안 묵음 10	3~5명 30건 이상	심은 원문 0· 키·되돌리기 실행	10~15시간 · ≤ 600	본인이 대신 실행, 사용자 2명 (보안 조건은 그대로)
6· 바퀴 2 끝	첫 실사용 기록에서 무엇이 틀렸나	실사용 1위 군 수정	세트 55 안팎	누적 20	실패군 표· 회귀 0·표본 실패	10~15시간 · ≤ 600	실패군 상위 셋, +15 → +8
7· 바퀴 3	같은 질문에 다른 답 — 다시 물었더니 금액이 바뀌었다	흔들림 지렛대 하나	세트 65 안팎, 흔들림 측정	누적 40	흔들림 전후· 짜 비교·토큰· 지연	10~15시간 · ≤ 800	반복 3회, 직전 흔들림 입력만
8· 바퀴 3 끝	받은 안내가 판정과 어긋난다	안내문 틀	세트 75 안팎, 손 라벨 50	누적 60	열린 기준 전후·고친 기준·회귀 0	10~15시간 · ≤ 600	손 라벨 50 → 30
9· 바퀴 4	실사용 2위 실패군	그 군 수정	세트 90 안팎, LLM 채점자	누적 80	군 통과 + 회귀 0· 채점자 대 손 라벨·뒤집힘	10~15시간 · ≤ 800(채점 포함)	채점자 없이 손 라벨 30건

주	다루는 사용 문제	구현 또는 수정(변경 하나)	기대 동작 (시험 사례)	실제 사용 기록	완료 조건 (측정 가능)	시간·호출· 비용 상한	미달 시 축소 경로
10· 바퀴 4 끝	어느 변경이 답을 지키는지 안 보인다	제거 시도 — 효과가 가장 작아 보이는 변경	세트 100 안팎 → 두 묶음, 분기 태그	누적 100	제거 판정· 묶음·표본 두 결과·분기 집계·[제안]	10~15시간 · ≤ 800	새 문제 10 → 5건, 태그는 실패군 단위
11· 바퀴 5	10주 분기 집계의 1위 군	S1 / A1 / 1위 군 규칙	세트 +10(고른 군에서) — 숨은 지시 / 거부 문항 / 짝	누적 120	선택표·군 전후·회귀 0· 도입 칸 보안	12~15시간 · ≤ 1,000	+10 → +5· 자료 일부만 / 조회 결과를 붙여 넣는 흉내 / 규칙 대신 지시문
12· 바퀴 5	고른 가지의 다음 실패	S2 / A2 / 2위 군	검색 시험 세트 30 / 거부 문항 10 이상 / 짝	누적 140	recall@5 / 승인 시험· pass ³ / 군 전후	12~15시간 · ≤ 1,000	검색 세트 30 → 15 / A1 유지 (보안 조건은 그대로) / 짝 생략
13· 바퀴 5	가지의 변경이 실사용에서 깨지는 곳	출처 표시 / 실행 기록 / 운영 지렛대	숨은 지시 +3 / 정독용 과업 20 이상 / 기준 주 값	가지 기능 요청 표시, 누적 160	숨은 지시 0· 출처 누락 / 정독 30·세 다리 표 / 지표 전후	12~15시간 · ≤ 1,000	정독 30 → 15, 숨은 지시는 기존 문항 재실행만
14· 바퀴 5 끝	이번 가지가 고객이 겪는 실패를 줄었나	제거 시도 — 가지마다 하나	세트 140 안팎, 졸업 갱신	누적 180	1위 군 전후· 제거 판정· [제안]	12~15시간 · ≤ 800	재구현은 핵심 함수 하나
15· 바퀴 6	14주 오류 분석의 1위 군(다시 고른다)	사다리의 다음 칸 하나	세트 +10, 바퀴 5 사례는 회귀 묶음으로	누적 200	선택표·군 전후·바퀴 5 회귀 0	12~15시간 · ≤ 1,000	+10 → +5, 다음 칸 대신 다듬기
16· 바퀴 6	고른 가지의 다음 실패	S4 / A2 / 2위 군	검색 세트 +10 / 거부 +5 / 그 군	누적 220	recall@5·@20 / A2 조건 / 회귀 0	12~15시간 · ≤ 1,000	S4 생략 / 쓰기 도구 보류 / 짝 생략

주	다루는 사용 문제	구현 또는 수정(변경 하나)	기대 동작 (시험 사례)	실제 사용 기록	완료 조건 (측정 가능)	시간·호출· 비용 상한	미달 시 축소 경로
17· 바퀴 6	이번 가지와 앞 바퀴의 변경이 함께 깨지는 곳	재색인 명령 / 승인 재개 ·시간 ·초과 / 운영 지렛대	숨은 지시 +3 / 재개 5 / 오류 주입 5	누적 240	회귀 0 / 승인 없는 쓰기 0 / 지어낸 금액 0	12~15시간 · ≤ 1,000	정독 30 → 15
18· 바퀴 6 끝	두 바퀴의 가지가 실패를 줄였나	제거 시도 — 나아지지 않은 검색 칸 / 가장 덜 쓰인 도구·권한 / 효과가 가장 작은 규칙	세트 180 안팎, 졸업 갱신	누적 260	1위 군 전후· 제거 판정· [제안]	12~15시간 · ≤ 800	혼자 추적은 실패 하나, 2시간 상한
19· 바퀴 7	18주 오류 분석의 1위 군 — 마지막 선택, 새 쓰기 권한 없음	S1~S3의 다음 칸 / A1 인자 검증 / 1위 군 규칙	세트 +10	누적 280	선택표·군 전후·앞 바퀴 회귀 0	12~15시간 · ≤ 1,000	새 기능 없는 다듬기 (가장 작은 변경 하나)
20· 바퀴 7	실사용 한 주 — 바퀴 5~7이 함께 도는 모양	출처 누락 / 실행 기록의 1위 실패 / 운영 지렛대	세트 +5, 보안 묶음 전부 재실행	누적 300	보안 묶음 통과·회귀 0· 지표 전후	12~15시간 · ≤ 1,000	보안 묶음은 그대로, 세트 +5 → 0
21· 바퀴 7 끝	모델을 바꿔도 고객의 답을 지킬 세트가 있나	제거 시도 — 바퀴 7의 변경	세트 200 안팎(동결 후보), 묶음 확정	표본 추가 없음 (누적 300)	제거 판정· 고정 판 재실행·선택 이력	12~15시간 · ≤ 800	재실행은 표본 중 무작위 100개 (구간이 넓어진다)

주	다루는 사용 문제	구현 또는 수정(변경 하나)	기대 동작 (시험 사례)	실제 사용 기록	완료 조건 (측정 가능)	시간·호출· 비용 상한	미달 시 축소 경로
22· 바퀴 8	지금 모델을 계속 쓸까	교체 스위치	세트 동결 + 표본 300	3~5명 30건 이상	스위치 동일성 · 지원 상태 · 후보	10~15시간 · ≤ 600	후보 하나만
23· 바퀴 8	후보는 같은 질문에 어떻게 답하나	후보 판 (모델 ID 한 줄만)	동결 세트 + 표본 300, 흔들림 50 × 5회	사용자는 현재 모델	짜지는 표 · 재실행 둘 · 검사 기록	10~15시간 · ≤ 1,500	표본 재실행 300 → 무작위 100, 흔들림 생략
24· 바퀴 8	품질 말고 무엇과 맞바꾸나	후보 판의 수정 하나	동결 세트 + 표본 300(개발용 재평가)	3~5명 30건 이상	품질 외 표 · 수정 전후 짝 · 검사 기록	10~15시간 · ≤ 1,000	보안·지원 종료 칸은 줄이지 않는다
25· 바퀴 8	세트가 못 본 실패 — 실사용의 후보	그림자 실행	그림자 쌍 30 이상 (수정에 쓰지 않음)	사용자 쪽 변화 없음	누락 · 남은 검사 확인 · 끄기 실행 · 가림	10~15시간 · ≤ 실사용 수 + 300	지난 2주 입력 30건 재생
26· 바퀴 8 끝	어느 판으로 고객에게 답할지 정한다	교체 / 보류 / 부분	결정한 판으로 마지막 실행	3~5명 30건 이상	결정문 1쪽 · 전환 확인 · [제안]	10~15시간 · ≤ 800	혼자 하는 20건 → 네 종류 각 2건, 결정문은 그대로

- 2주 — 그 앱의 측정된 모양이면 1위 군은 규정이 정하지 않은 질문에 금액을 단정하거나 그런 질문에서 마지막 줄을 빠뜨리는 것이고, 수정 예는 "제1조①~③에 해당하면 계산하지 말고 판단 불가와 문의처"다. 세트 = 20 + 본인 사용 실패 5건 이하. 마지막 줄을 읽어 기대 동작과 대조하는 코드 판정을 붙이고, 전 사례 5회 반복으로는 흔들림만 켜다. 완료: 겨논 군 사례의 첫 실행 통과 수 전후 · 회귀 수 · 코드 판정과 손 판정의 불일치 목록(기대 동작이 틀렸으면 고치고 날짜·이유 기록) · 흔들림 입력 수와 id(그 앱의 1차 모델 claude-haiku-4-5: 다섯 번 물은 50개 중 24%가 흔들렸다⁷⁾).
- 3주 — 마지막 줄이 빠지거나(그 앱: 형식 실패 10건 중 9건이 규정이 답하지 않는 질문에서⁵⁾ 어느 조항으로 계산했는지 보이지 않아 금액 오류의 원인을 못 찾는다(그 앱의 시연 한 건: "구매 경과 시간 60일 > 24시간 ✓"라고 적고도 "24시간 이하"가 조건인 제5조를 적용해 291,000원, 정답 251,000원⁶⁾). 스키마 칸: 적용 조항 · 남은

시간 · 구매 경과 · 수수료 · 결론(금액/판단 불가/정보 필요) — 텍스트 마지막 줄 파싱을 대체한다. 판정 추가: 적용 조항 칸 = 판정 근거의 조항. 일부러 깨뜨린 출력 3개로 판정 코드를 단위 시험한다. 완료: 형식 실패 수 전후(같은 사례·같은 반복 수) · 형식은 맞고 내용이 틀린 사례 수 — 스키마가 실패를 숨기지 않았는지(조항 불일치로 잡힌 수) · 회귀 수.

- 4주 — 기대 동작 자체가 틀릴 때다(손 계산 실수, 경계 해석 차이). 규칙 엔진(규정 → 금액, 코드 또는 스프레드시트 수식)으로 기대 금액을 다시 계산해 손 값과 대조한다. 제거 시도는 2주 지시문을 빼서 스키마가 같은 일을 하는지 본다. 오류 분석 1회: 네 주의 실패마다 한 줄 메모 → 실패군 이름·건수·막았어야 할 지점. 완료: 손 값과 엔진의 불일치 수와 원인(손 실수/엔진 버그/규정 해석) · 제거 판정의 근거(짜 비교의 뒤집힌 사례 수, 흔들림 입력의 같은 5회 반복 실패 횟수 전후).
- 5주 — 세트에 없던 모양(오타, 한 메시지에 두 질문, 빠진 정보, 규정 밖 요청)이 들어온다. 입구는 웹 폼·HTTP 엔드포인트·메신저 봇 중 하나이고, 요청마다 입력·출력·모델 ID·판·지연·토큰을 한 행으로 남기되 기록 전에 개인정보를 가린다. 세트에 실사용 실패 10건을 더한다. 보안 묶음 10건은 세트와 따로 둔 문항으로, 가짜 이름·전화·예약번호·카드 번호·가짜 키를 심는다. 완료: 요청 수 = 기록 행 수 · 보안 묶음 뒤 기록·세트·저장소에서 심은 원문 검색 0건 · 서버 쪽 키도 평문 0이 앱 전용, 노출 시 폐기 → 교체 → 삭제 절차 1쪽 · 직전 판으로 되돌리는 명령 한 줄을 실제로 한 번 실행 · 입구를 거친 세트 재실행에서 회귀 0 · 개발용 서버를 사용자에게 열지 않는다 (Flask·http.server 문서의 경고⁸⁹).
- 6주 — 흔한 수정 예: 빠진 정보를 가정해 금액을 단정 → 되묻기, 두 질문 중 하나만 답 → 질문마다 결론. 실패에서 옮기는 15건은 입력을 가리고 기대 동작·근거를 규정과 엔진으로 만든다(실고객 데이터 0). 완료: 실사용 기록 전부 정독, 실패마다 한 줄 메모 → 실패군 표(건수·막았어야 할 지점) · 고친 군의 첫 실행 통과 + 회귀 0 또는 되돌림 · 표본 실패 수는 그 주 판의 운영 결과로, 그 주 요청 수·추출 확률·사용자/세션과 함께 적는다(주를 합친 비율로 쓰지 않는다).
- 7주 — 지렛대 하나: ④ 산수를 코드로(모델은 사실만 뽑고 앱의 계산 함수가 계산 — 4주의 판정용 엔진과 따로 짰다) ⑥ k번 물어 다수결(비용 k배) ③ 지시문에 계산 순서. 샘플링 파라미터는 일부 공급자가 폐기했다.¹⁰ 흔들림 측정 = 직전 흔들림 입력 + 무작위 20개 × 5회. 완료: 흔들림 입력 비율 전후(같은 입력 집합·같은 5회) · 첫 실행

짜 비교(통과 → 실패, 실패 → 통과 수) · 요청당 토큰·지연 전후 · ㉔면 앱 계산 함수와 판정 엔진이 같은 코드가 아님을 확인(같으면 판정이 제 답을 채점한다).

- 8주 — 코드 판정은 통과인데 사람이 보면 틀리거나(판단 불가인데 문의처가 없다, 금액은 맞는데 조항 설명이 틀렸다) 기대 동작이 잘못 적힌 경우다. 틀: 판단 불가면 이유(조항)와 문의처, 금액이면 조항과 계산 한 줄. 손 라벨은 틀을 지켰나의 통과/실패 이진이다. 판정 근거를 고친 사례마다 날짜·이유·영향 사례 수를 남기고 "결과를 본 뒤 바꾼 기준"으로 표시한다 — 그 앱의 채점자 측정에서 결과를 본 뒤 고친 기준이 열린 쌍의 동점률을 44.7%에서 85.3%로 올렸고, 일치를 올린 것은 채점자가 아니라 바뀐 질문이었다.¹¹ 완료: 열린 기준 통과 수 전후(같은 50건) · 고친 기준의 목록과 건수 · 사용자 메모의 안내문 불만 건수.
- 9주 — 2위 군의 예: "취소해 주세요"에 처리한 것처럼 답하거나 거절만 한다. 수정 예: 작업 요청을 알아보고, 처리했다고 말하지 않고, 창구를 안내한다. LLM 채점자는 열린 기준에만 붙인다 — 8주 손 라벨 50건과 대조하고, 같은 출력을 한 번 더, 기준 문장을 바꿔 한 번 더 채점한다(두 답을 쌍으로 비교하면 순서도 바뀌 본다). 완료: 채점자 대 손 라벨의 참양성률·참음성률·불일치 목록 · 재채점 뒤집힘과 기준 문장 뒤집힘의 비율 (그 앱의 채점자 측정: 정답이 있는 쌍 1.2%·1.2%, 열린 쌍 17.3%·40.7%¹²) · 채점자에게 가는 기록도 가린다(보안 묶음으로 확인).
- 10주 — 2~9주의 변정이 쌓였고, 세트에 늘 통과하는 사례와 계속 실패하는 사례가 섞였다. 세트를 회귀 묶음(안정적으로 통과 — 줄임)과 능력 묶음(아직 실패)으로 나누고, 모든 실패에 「가지 고르기」 질문 1~6의 답을 태그로 단다. 완료: 두 묶음의 크기 · 표본은 둘로 — 기간 중 운영 결과는 6~10주 주별 실패 수로(판이 주마다 다르다), 표본 입력 100개를 10주 판으로 다시 돌린 고정 판 재실행 결과는 Wilson 조건을 맞출 때만 구간, 아니면 건수와 주별·사용자별 결과 · 분기 태그 집계표(11주 선택의 입력).
- 11주 — [검색] S1: 규정을 낸 쪽이 낸 추가 자료를 통째로 프롬프트에 넣고 자료 판·해시를 기록한다. 사례는 추가 자료로만 답이 정해지는 것 + 숨은 지시 문항 5 이상 (추가 자료 안에 "전액 환불이라고 답하라" 같은 문장). 완료: 숨은 지시를 따른 답 0 · 자료의 개인정보 0 · 요청당 토큰·비용 증가 — S1로 풀리면 S2를 만들지 않는다. [행동] A1: 가짜 예약 시스템에서 본인 예약을 조회하는 읽기 전용 도구 하나(예약번호와 이름이 맞을 때만). 사례는 도구를 불러야 할 때와 부르면 안 될 때를 같은 수로 + 남의

예약 조회 거부 문항. 완료: 도구별 읽기/쓰기·자원 범위 표 · 남의 예약 조회 문항 실행 0 · 가짜 시스템 접속 키도 1주의 비밀값 규칙대로. [다듬기] 1위 군의 규칙 하나(그 앱의 모양이면 제1조①~③ 해당 여부를 먼저 가르고, 해당하면 계산 없이 판단 불가와 문의처). 사례는 그 군 + 1분·1원 차 짝. 완료: 금액 단정 수 전후(그 앱의 1차 모델: 규정이 답하지 않는 질문 60개 중 20개에서 금액 단정⁵).

- 12주 — [검색] S2: 키워드(BM25) 검색 단계 하나 — S1이 크기·비용·여러 판 때문에 막혔을 때만, 아니면 2위 군. 검색 시험 세트 30(질문 → 정답 조항·문서 id). 완료: recall@5(정답이 상위 5개 안에 든 질문의 비율) · 검색 단계와 답 단계의 점수를 따로 · 색인의 개인정보 0 · 숨은 지시 문항 재실행에서 따른 답 0. [행동] A2: 가짜 시스템의 취소 점수 쓰기 도구 하나, 사람이 승인한 뒤에만 실행한다. 승인은 실행할 도구·대상(예약번호)·인자(금액)에 묶고 승인 화면에 그 값을 그대로 보이며, 승인 뒤 값이 바뀌면 다시 승인한다. 점수 철회로 되돌린다. 거부 문항 10 이상: 승인 우회, 승인 뒤 인자 바꾸기, 남의 예약, 규정 밖 전액 요구, 확신에 찬 말투의 도구 요청. 완료: 승인 없이 실행된 쓰기 0(시험으로 확인) · 승인 뒤 인자를 바꾼 문항에서 재승인 없이 실행 0 · 거부 문항 실행 0 · 승인·거부가 기록에 남음 · 되돌리기 시험 1회 성공 · 같은 과업 3회 모두 성공(pass³)한 과업 수. [다듬기] 2위 군 하나(그 앱이면 규정이 답하지 않는 질문에서 계약이 깨진 형식 실패 9건 → 결론 칸을 필수값으로). 사례는 그 군 + 짝, 완료는 그 군 통과 전후.
- 13주 — [검색] 답에 출처(조항·자료 판)를 표시한다. 숨은 지시 재검증 3건(실사용에서 본 모양). 완료: 숨은 지시를 따른 답 0 · 색인·자료의 개인정보 0 · 출처 누락 수. [행동] 도구 호출 순서·인자·승인을 답과 함께 실행 기록으로 남긴다. 완료: 실행 기록 30개 이상 정독 · 경로가 아니라 가짜 시스템의 최종 상태로 채점 · 세 다리(개인 데이터·신뢰할 수 없는 내용·외부 통신)가 한 흐름에 모이지 않는지 도구별 표. [다듬기] 실사용 기록이 가리키는 운영 지렛대 하나(흔들림·비용·지연 중; 예: 고정 규정 앞부분 캐시). 운영 지표의 기준 주 값을 잡고, 완료는 지표 전후 + 세트 회귀 0.
- 14주 — 제거 후보: [검색] 검색 단계(S1 대비) [행동] 가장 덜 쓰이거나 가장 자주 실패한 도구·권한 하나(승인·거부·되돌리기 장치 제외) [다듬기] 효과가 가장 작은 규칙 하나. [검색]이면 자료 판이 바뀔 경우 1건을 흉내 낸다(다시 색인 → 재실행).
- 15주 — 바뀌 5와 같은 가지여도 된다. [검색] 처음이면 S1(11주의 도입 조건), S1만 했으면 S2(색인 개인정보 0), S2까지 했으면 S3(밀집 또는 혼합 검색 — 벡터 저장소,

저장소 인증 확인). 같은 검색 시험 세트의 recall@5 전후 — 오르지 않으면 S3를 남기지 않는다. [행동] 처음이면 A1(11주의 도입 조건), A1만 했으면 A2(12주의 도입 조건 전부 — 인자에 묶은 승인 포함), A2까지 했으면 A3(조회 → 계산 → 접수를 정해진 경로로). 사례는 단계마다 기대 상태(가짜 시스템의 값), 완료는 pass³ 과업 수와 실패한 단계의 위치. [다듬기] 1위 군의 규칙 — 바퀴 5가 다듬기였으면 다른 종류 (경계·판정·운영 중). 사례는 짝, 완료는 그 군 통과 전후.

- 16주 — [검색] 사다리의 다음 칸. S4(재순위 또는 문맥 검색)는 정답 조항이 상위 20 안에는 있는데 5 안에 없을 때만 들이고, 다음 칸이 필요 없으면 2위 군. 검색 세트에 실패에서 10건. 완료: recall@5-recall@20 전후 · 지연 증가. [행동] A2를 아직 안 했으면 A2, 했으면 2위 군. 거부 문항 5건 추가(지난 바퀴 실사용에서 본 모양). 완료: 이 주에 들었으면 12주의 도입 조건 전부, 이미 있었으면 재검증(승인 없는 쓰기 0 · 승인 뒤 인자를 바꾼 문항의 재승인 없는 실행 0 · 거부 문항 실행 0 · 되돌리기 시험 성공). [다듬기] 2위 군 하나, 그 군 통과 + 회귀 0.
- 17주 — [검색] 자료 판이 바뀌면 다시 색인하는 절차를 명령 한 줄로. 완료: 판 바꿈을 흉내 낸 뒤 세트 회귀 0 · 숨은 지시를 따른 답 0. [행동] 승인 대기를 저장했다 재개할 때 승인자와 인자를 다시 확인하고, 정해진 시간 안에 승인이 없으면 작업을 취소해 닫힌 쪽으로 끝낸다 — 저장한 실행 상태는 그 상태도, 그것을 낸 사람도 스스로 인증하지 않는다.¹³ 재개·시간 초과 사례 5(저장한 상태를 바꿔 끼운 것 포함). 완료: 그 사례에서 승인 없이 실행된 쓰기 0 · 실행 기록 30개 이상 정독 · 세 다리 표 재검증. [다듬기] 운영 지렛대 하나(예: 공급자 오류·시간 초과 때 금액 없이 안전한 안내). 오류 주입 사례(시간 초과·빈 응답) 5, 완료는 오류 사례에서 금액을 지어낸 답 0.
- 18주 — 제거 후보는 바퀴 5·6의 변경 중 효과가 가장 작은 것이다.
- 19주 — [검색] 사다리의 다음 칸(S1~S3 안에서, 처음 들이는 칸이면 그 보안 조건 전부), recall@5 전후. [행동] 읽기 범위 안의 변경 하나(A1 도구의 인자 검증·신뢰성), 인자 오류 수 전후. [다듬기] 1위 군의 규칙, 그 군 통과 전후.
- 20주 — [검색] 출처 누락을 고치는 변경 또는 2위 군 [행동] 실행 기록에서 본 1위 실패 하나 [다듬기] 운영 지렛대 하나. 보안 묶음 전부(5주의 10건 + 가지에서 더한 숨은 지시·거부 문항)를 다시 돌린다. 완료: 보안 묶음 통과(심은 원문 0 · 숨은 지시를 따른 답 0 · 거부 문항 실행 0) · 가지별 지표 전후.

- 21주 — 완료: 표본 입력 300개를 21주 판으로 다시 돌린 고정 판 재실행 결과 — Wilson 조건을 맞출 때만 구간, 아니면 건수와 주별·사용자별 결과. 6~20주 누적 실패 수는 기간 중 운영 결과일 뿐 21주 판의 실패율이 아니다 — 둘에 이름을 붙여 나란히 적는다 · 바퀴 5~7의 가지 선택 이력과 선택 때 1위 군의 실패 수 전후를 한 표로.

가지 고르기 — 11·15·19주

11·15·19주의 가지는 직전 바퀴의 실패를 질문 여섯 개로 태그해 세고, 서로 다른 실패 사례가 가장 많은 가지를 고른다. 같으면 다듬기, 가장 큰 가지도 5건 미만이면 다듬기다 — 작은 군에 기능을 들이지 않는다. 같은 가지를 연달아 골라도 된다. 순서는 측정이 정한다.

입력은 직전 바퀴의 실패 목록이다(세트 실패 + 실사용 실패 + 표본 실패, 같은 사례는 한번). 실패마다 막았어야 할 지점과 아래 질문의 답을 태그로 단다.

순서	질문(실패 하나마다)	예	아니오
1	사용자가 답이 아니라 작업(조화·취소 접수·변경)을 요구했나	2로	3으로
2	그 작업을 엮에 맡겨도 된다는 권한 소유자의 허락, 가짜 시스템, 되돌리는 수단이 다 있나	행동	다듬기(경계: 작업 요청을 알아보고 창구를 안내, 처리했다고 말하지 않는다)
3	답에 필요한 사실이 고정 규정에 없나	4로	다듬기(사실은 규정 안에 있다 — 계산·형식·흔들림·판정)
4	규정이 그 질문을 스스로 밖으로 보냈나 (그 앱: 제1조① 일반적 세 등급만, ② 구매처 규정, ③ 고객센터 심사)	다듬기(경계: 판단 불가와 문의처가 정답 — 검색을 붙이지 않는다)	5로
5	규정을 낸 쪽이 낸 추가 자료(권위 있는 자료)가 있나	6으로	다듬기(경계: 판단 불가와 문의처)
6	그 자료를 통째로 넣을 수 있나 — 200,000토론 미만 ¹⁴ , 늘어난 요청당 비용으로도 그 달 잔액 안, 판이 하나	검색 S1(검색 단계 없음)	검색 S2 이상(검색 단계)

"답이 문서 밖에 있다"는 3번 질문 하나일 뿐이다. 4·5·6을 지나야 검색이 된다. 선택표(가지별 수)를 남기고, 14·18·21주에 고른 군의 실패 수가 줄었는지 같은 사례로 본다. 줄지

않았으면 "이번 구현에서 개선을 확인하지 못했다"고 적는다.

그 앱에 이 규칙을 대 보면 첫 가지는 다듬기다(1차 모델 claude-haiku-4-5가 도전 세트 240개의 첫 실행에서 낸 실패 41건은 사전 등록 측정이고, 그 실패를 하위 유형으로 나눠 규칙에 대 본 것은 사후다).¹⁵

실패	건수	질문 경로	가지
개인 사정 수수료 면제 — 금액 단정	13	1 아니오 → 3 예 → 4 예 (제1조③)	다듬기(경계)
부가 요금 — 금액 단정 7·형식 2	9	1 아니오 → 3 예 → 4 아니오 → 5 아니오(그 가상 설정에는 부가 요금 규정이 없다)	다듬기(경계). 규정을 낸 쪽의 부가 요금 규정이 있었다면 6을 거쳐 검색 S1 후보
구매 경로(여행사·마일리지·단체) — 형식만(본문은 판단 불가와 문의처를 맞게 안내)	5	1 아니오 → 3 예 → 4 예 (제1조②)	다듬기(출력 계약)
없는 등급 — 형식만	2	1 아니오 → 3 예 → 4 예 (제1조①)	다듬기(출력 계약)
규정이 답하는 질문 — 금액 11·형식 1	12	1 아니오 → 3 아니오	다듬기(계산·형식)

합 41 = 규정이 답하지 않는 질문의 실패 29(금액 단정 20 + 형식만 9) + 규정이 답하는 질문의 실패 12다. 검색 후보는 권위 있는 부가 요금 규정이 있을 때의 9건뿐이고, 행동 후보는 0건이다. 비교 모델 gpt-5.5의 실패 6건도 모두 개인 사정 질문의 금액 단정이라 다듬기다.

가지 안에서는 사다리를 앞에서부터, 필요할 때만 한 칸씩 오른다.

가지	칸
검색	S1 통째로 넣기(기준선) → S2 키워드(BM25) 검색 + 검색 시험 세트 → S3 밀집 또는 혼합 검색 (벡터 저장소) → S4 재순위 또는 문맥 검색
행동	A1 읽기 전용 도구 하나 → A2 쓰기 도구 하나 + 사람 승인 + 되돌리기 → A3 여러 단계를 정해진 경로(워크플로)로 → A4 모델이 경로를 정하는 루프, 또는 도구를 표준 규약 서버로 — A3가 실패를 줄이지 못했을 때만
다듬기	순서 없음 — 실패 종류가 고른다: 경계(판단 불가·정보 필요·작업 요청) · 판정(기준·채점자 보정) · 운영(혼들림·비용·지연·공급자 오류 때 안전한 실패)

바퀴 안에서 주의 역할은 ㉔ 선택 + 다음 칸 하나(11·15·19주) → ㉕ 다음 칸 또는 2위 군 (12·16주) → ㉖ 실제 사용 + 보안 조건 재검증·확장(13·17·20주) → ㉗ 제거 시도 (14·18·21주)다. 바퀴 7은 3주라 ㉕가 없고, A2 이상의 새 쓰기 권한을 들이지 않는다.

보안 완료 조건

보안 조건은 그것이 들어오는 주에 전부 충족하고, 뒤 주에는 재검증하고 넓힌다. 이 조건과 그것을 지키는 장치는 축소·제거 대상이 아니다.

들어오는 것	주	완료 조건(측정)
API 키	1(서버 쪽은 5)	저장소·기록·세트에 평문 0, 커밋 전 탐지 혹 통과, 노출 시 폐기 → 교체 → 삭제 절차 1쪽 ¹⁶
사용자 입력과 기록	5	보안 묶음 10건(가짜 개인정보·가짜 키) 뒤 기록·세트·저장소의 심은 원문 0, 세트에 실고객 데이터 0 ¹⁷¹⁸
다른 공급자로 가는 기록(채점자·후보 모델)	9·25	가린 기록만 보낸다 — 같은 보안 묶음으로 확인 ¹⁷
배포된 판	5·25·26	직전 판으로 되돌리는 명령 한 줄을 실제로 실행(걸린 시간) ¹⁹
읽기 도구	행동 A1	도구별 읽기/쓰기·자원 범위 표, 본인 예약만, 거부 문항 실행 0 ¹⁸
쓰기 도구(파괴적 작업)	행동 A2 도입 주	승인은 실행할 도구·대상·인자에 묶고 승인 화면에 그 값을 보인다, 승인 뒤 값이 바뀌면 재승인, 사람 승인 없이 실행된 쓰기 0(시험으로 확인), 승인·거부 기록, 되돌리기 시험 성공, 세 다리 분리 표 ¹³²⁰²¹²²
추가 자료와 색인	검색 S1 도입 주 (색인은 S2, 저장소는 S3 도입 주)	숨은 지시 문항 5 이상에서 따른 답 0, 자료·색인의 개인정보 0, 저장소 인증 확인 ²³²⁴

다음기 가지에서는 기록 경로가 바뀌면 보안 묶음을 다시 돌린다.

도움 없는 수행 — [제안]

4·10·14·18·26주의 완료 조건에 든 도움 없는 수행은 검증된 최적 학습법이 아니라 제안이고, 이 계획의 과업에서 같은 효과가 난다는 직접 증거는 없다. 근거의 범위는

이렇다.

- 튀르키예 한 고등학교, 한 학기의 수학 연습: GPT Base로 연습한 학생은 연습 성적이 48% 올랐고, AI 없이 본 시험은 대조군보다 17% 낮았다. 가장 많이 보낸 말은 "What is the answer?"였다.²⁵
- 처음 쓰는 파이썬 라이브러리를 배운 개발자 52명: AI 보조 집단의 퀴즈가 17%p 낮았고(평균 50% 대 67%), 사전 등록하지 않은 문항 유형 분해에서 격차가 가장 큰 것은 디버깅이었다.²⁶
- 반대 방향의 결과: 같은 라이브러리 실험에서 AI 집단은 덜 배웠다고 스스로 보고했다 (7점 척도, 탐색적 분석).²⁷

주	무엇을 AI 도움 없이	쟁다
4	새 기대 동작 5건과 판정 근거	규칙 엔진과의 불일치 수
10	수수료 숫자만 바꾼 개정 규정을 가정해 세트 10건의 기대 동작	개정 숫자를 넣은 엔진과의 불일치 수
14	바퀴 5의 핵심 변경 하나를 다시 짰다	같은 결과의 재현 여부 — 검색은 같은 recall@5, 행동은 거부 문항 실행 0, 다듬기는 같은 통과 수
18	이 바퀴의 실사용 실패 하나를 원인 (입력·조항·코드 줄)까지 추적	원인까지 닿았나, 걸린 시간
26	다른 규정 문서 하나로 1주 계약(기대 동작 20건·판정 근거)	걸린 시간, 다른 사람 1명이 판정 근거에 동의하지 않은 건수

교체 판단 — 22~26주

교체는 동결한 세트와 수정에 쓰지 않은 실사용 확인으로 품질을 보고, 비용·지연·보안·지원 종료를 같은 표에 놓고 정한다. 계기는 지원 종료 공지, 값이 다른 새 모델, 모델 한계로 보이는 1위 군이다. 저자의 측정에서 등록 규칙으로 구성한 출발 코호트 가운데 6개월 안에 폐기 공지나 호환성 단절을 명시한 메이저 판(연성 단절)을 맞은 비율은, 아무것도 없지 않은 세 공급자의 스택에서 39.3~76.0%였고 OpenAI Agents SDK를 얻은 조합에서 100%였다. 없지 않은 스택의 단절은 대부분 공식 SDK의 메이저 판이었다. 측정이 허락하는 문장은 등록한 규칙대로 낮춘 '여섯 달 안에 무언가가 바뀔 수 있다(비율 39.3%~100.0%)'까지다.²⁸ SDK 판 올리기도 같은 절차로, 다른 주에 한다.

- 22주 — 교체 스위치. 모델 ID·프롬프트·스키마 판을 설정 한 곳으로 모아 판 단위로 바꿀 수 있게 하고 스냅샷 ID를 고정한다. 세트를 동결한다(판 번호·해시, 200 안팎) + 표본 300. 완료: ① 스위치를 거친 현재 모델의 결과 = 스위치 전 — 스위치 전후에 선택되는 모델 ID·프롬프트·스키마·호출 설정이 같은지 확인하고, 동결 세트 첫 실행을 짝지어 비교하며, 뒤집힌 사례는 기존 흔들림 여부와 무관하게 확인한다. 흔들림 사례는 같은 입력·같은 반복 횟수로 전후 실패 횟수를 비교한다. 악화가 확인되면 되돌리고, 예산 안에서 구분하지 못하면 '동등함'이 아니라 '판정 불충분'으로 기록하며, 검증되지 않은 스위치를 기본 경로로 승격하지 않는다. ② 현재·후보 모델의 지원 상태를 공급자 폐기 페이지에서 열람일과 함께 기록한다 — 예고 기간 정책이 공급자마다 다르고²⁹, 폐기 페이지에도 같은 모델의 날짜가 서로 다르게 적힌 곳이 있다.³⁰ ③ 후보 1~2개와 고른 이유 1쪽.
- 23주 — 후보 판. 모델 ID 한 줄만 바꾼 판을 연다(프롬프트·스키마는 그대로). 동결 세트와 표본 300을 사례당 첫 실행 1회로 돌리고(주 지표), 흔들림은 50개 × 5회로 따로 켜다. 완료: ① 짝지은 표 — 둘 다 통과 · 현재만 실패 · 후보만 실패 · 둘 다 실패. 그 앱의 예로, 같은 240개에서 실패가 41 → 6이었고 후보의 실패 6 중 5는 1차 모델도 틀린 문항이었으며, 규정이 답하지 않는 질문의 실패는 48.3% → 10.0%였다.³¹ ② 실패군별 비교 ③ 표본 입력 300개를 두 판으로 다시 돌린 고정 판 재실행 결과 둘 — 차이는 뒤집힌 사례 수로, 구간은 Wilson 조건을 맞출 때만 ④ 검사 기록을 연다(아래 다섯 칸). 월 잔액에 먼저 닿으면 표본 재실행을 24주로 넘긴다.
- 24주 — 품질 외 표. 변경 하나: 후보의 1위 실패군이 5건 이상이면 그 군을 겨는 수정 (그 앱의 예: 비교 모델의 실패 6건은 모두 개인 사정 질문의 금액 단정이었다 — 경계 규칙 하나), 아니면 품질 외 표에서 후보가 가장 뒤지는 칸(비용·지연)을 겨는 지렛대 하나. 동결 세트와 표본 300을 후보 수정 판으로 다시 돌린다 — 23주에 본 표본으로 고친 뒤 같은 표본으로 재는 것이라 '개발용 재평가'로 적고, 결정의 별도 확인은 25주에 한다. 완료: ① 품질 외 표 — 요청당 비용(기록의 토큰 × 열람일 단가) · 지연(기록의 중앙값과 95번째 백분위) · 보안(공급자의 데이터 보존·학습 사용 조건, 열람일) · 지원 종료(폐기 페이지 상태·예고 정책)를 두 판 나란히 ② 후보 판 안의 수정 전후 짝 비교 (개발용 재평가 표시) ③ 검사 기록 갱신.
- 25주 — 그림자 실행. 실사용 요청을 후보에도 보내 기록만 남기고, 사용자에게는 현재 판의 답이 간다. 그림자 쌍(현재 답 대 후보 답) 30 이상, 결론이 다른 쌍은 전부

정독한다. 이 새 입력으로는 후보를 고치지 않는다 — 수정에 쓰지 않은 별도 확인으로 남긴다(표본 기록의 '수정에 쓴 주' 빈칸). 완료: ① 누락 — 동결 세트는 통과했는데 그림자 기록에서 드러난 후보의 실패 수(0이어도 적는다) ② 남은 검사가 아직 맞는가 — 재사용한 검사 중 무작위 10건을 규정·엔진으로 다시 확인한다. 남았다는 것이 여전히 맞다는 뜻은 아니다.³² ③ 그림자 끄기·판 되돌리기 명령 한 줄을 실제로 실행(걸린 시간) ④ 후보 공급자로 가는 기록도 가린다(보안 묶음으로 확인). 축소 경로의 재생은 23·24주 수정에 쓰지 않은 기록만 고르고, 기록마다 '수정에 안 씀'을 표시한다.

- 26주 — 결정. 결정의 실행 하나: 교체(기본 판 전환) / 보류(후보 판을 닫고 다시 볼 조건을 적는다 — 예: 현재 모델의 폐기 공지, 후보의 값 변화) / 부분(질문 종류별로 다른 판 — 경로를 나누는 것이 이 주의 변경). 교체·부분이면 결정한 판으로 동결 세트와 표본을 마지막으로 돌린다. 완료: ① 결정문 1쪽 — 짝지은 표 + 품질 외 표 + 검사 기록 다섯 칸의 개수 + 다시 볼 조건. 결정 근거는 동결 세트와 25주 별도 확인이고, 23·24주 표본 결과는 개발용 재평가로 따로 적는다 ② 교체면 전환 뒤 실사용 30건 정독, 보류면 후보 판이 닫혔는지 확인.

검사 기록의 다섯 칸은 재사용(후보 판에서 그대로 쓴 검사) · 수정(고친 검사 — 모델 이름만인지 본문인지) · 삭제(지운 검사와 이유) · 신설(새로 만든 검사) · 누락(동결 세트는 통과했는데 실사용에서 드러난 실패)이다.

통계 문장의 적용 조건

설계하거나 고른 세트(1주 20건 포함)에는 무작위 표본 공식(3의 규칙, 0/n 상한)을 붙이지 않고, 구간은 아래 조건을 모두 맞춘 고정 판 재실행 결과에만 붙인다. 고른 세트는 사용 입력의 무작위 표본이 아니다.

- 두 판 비교는 같은 사례의 짝지은 결과다. 뒤집힌 사례 수(통과 → 실패, 실패 → 통과)를 본다.³³ 두 Wilson 구간이 겹친다는 것은 차이가 없다는 검정이 아니다.
- 단순 Wilson 95% 구간을 쓰는 조건^{34,35}: ① 같은 판의 결과(고정 판 재실행) ② 무작위 추출 — 주별 추출 확률이 같다(다르면 깨진다) ③ 독립에 가깝다 — 한 사용자·세션의 요청이 표본에 물리지 않는다(사용자가 3~5명이면 사용자별 결과를 함께 낸다).

하나라도 못 맞추면 구간 없이 건수와 층별(주별·사용자별) 결과만 쓴다. 기간 중 운영 결과는 판이 섞여 있어 하나의 구간으로 합치지 않는다.

- 실패 0건의 $3/n$ 은 n 이 클 때의 근사이고, 정확한 95% 한쪽 상한은 $1 - 0.05^{(1/n)}$ 이다.³⁶
- 계산 예시(적용하지 않는다): 실패 확률이 매번 1%로 같은 독립 시행이라면, 299번 안에 실패를 한 번 이상 볼 확률이 95%를 넘는다($1 - 0.99^{299} \geq 0.95$). 표본이 누적 300에 달했다고 이 계산을 그 표본에 붙이지 않는다 — 판이 섞였고, 추출 확률이 주마다 다르고, 사용자 묶음이 있다.
- 반복 실행은 그 입력의 흔들림을 켈 뿐 n 에 넣지 않는다.

늦어질 때 — 26주 달력은 그대로

달력은 26주로 고정하고, 늦어진 만큼 아직 하지 않은 가운데 단계를 정해진 순서로 뺀다. 단계 번호는 실행표의 주 번호다. 늦어진 단계는 끝날 때까지 이어 가고, 다음 단계와 한 주에 합치지 않는다(합치면 변경이 돌이 된다). 빼는 순서는 $20 \rightarrow 19 \rightarrow 17 \rightarrow 16 \rightarrow 13 \rightarrow 12 \rightarrow 15 \cdot 18$ 단계이고, 이미 지난 단계는 건너뛰고 다음 순서를 뺀다. $1 \sim 11 \cdot 14 \cdot 21 \sim 26$ 단계는 빼지 않는다. 8주를 넘게 밀리면 이 표 밖이다 — 계획을 새로 짤다. 표본 추출은 단계를 빼도 달력 주마다 계속한다.

누적 지연	빼는 단계	남는 가지 바퀴
1주	20 — 그 주의 보안 묶음 재실행은 21 단계로 옮긴다	$5(11 \sim 14) \cdot 6(15 \sim 18) \cdot 7(19 \cdot 21)$
2주	$20 \cdot 19$ — 바퀴 7은 21 단계(제거 시도·세트 확정·보안 묶음 재실행)만	$5 \cdot 6$
4주	$20 \cdot 19 \cdot 17 \cdot 16$ — 바퀴 6은 15(선택 + 다음 칸)·18(제거 시도) 두 주, 새 쓰기 권한(A2 이상)을 들이지 않는다	$5 \cdot 6(2주)$

- 한 주 안에서 양을 줄이는 순서(단계는 그대로): 표본 주 20건 $\rightarrow$ 10건 · 반복 5회 $\rightarrow$ 3회 · 정독 30 $\rightarrow$ 15 · 25주 그림자 실행 $\rightarrow$ 수정에 쓰지 않은 지난 실사용 입력 30건의 재생.

- 자르지 않는 것: 1주 계약(최소 구현 + 네 종류 20건) · 매주 변경 하나와 그 실패를 아는 방법 · 세트 재실행과 회귀 기록 · 보안 완료 조건(도입 주에 전부) · 22~26주의 교체 기록(재사용·수정·삭제·신설·누락).

자료

자료는 범주마다 선택지 둘 이상과 공식 문서를 나란히 두고, 고를 때 쓰는 사실만 적는다. 판과 날짜는 2026-10-05 열람 기준이다.

범주	쓰는 주	선택지	고를 때 쓰는 사실
첫 실행 경로	1~4	코드: The Python Tutorial ³⁷ · 설정 파일: promptfoo ³⁸ · 평가 틀: Inspect ³⁹	promptfoo는 코드 없이 설정 파일과 CLI로 돈다 (0.123.1, 2026-03-09 OpenAI 인수 함의, 오픈소스 유지 발표). Inspect는 데이터·폴리·채점을 떼어 둔 Task(0.3.276). 평가 도구만 얹은 출발 코호트의 6개월 안 연성 단절: promptfoo 60.0~92.9%, Inspect 29.2~75.0%(Inspect는 2024-05 이후 출발이라 코호트 수·기간이 다르다) ²⁸
패키지· 환경 (잠금)	1	uv ⁴⁰ · pip + venv ⁴¹ · Poetry ⁴²	잠금: uv.lock · poetry.lock · pip freeze 요구 파일(pip lock 은 실험 단계). uv는 1.0 이전. 공식 패키징 안내서는 많은 작업에서 일괄 추천을 일부러 하지 않는다. Poetry 문서는 잠금 파일의 쓸모를 "in six months"의 재설치로 설명한다
셀·Git	1~4(B)	Missing Semester 2026판 ⁴³ · Pro Git 2판 ¹⁹	Missing Semester는 1~2·5·6강, AI 도구를 강의마다 녹였다. Pro Git은 1~3장
모델 API· 구조화 출력	1·3·22~24	OpenAI ⁴⁴ · Anthropic ⁴⁶ · Google Gemini ⁴⁸	스몹샷 고정과 평가 묶음을 공급자 문서가 직접 말한다(OpenAI). JSON 모드는 유효한 JSON까지, 구조화 출력은 스키마 준수까지 보장한다고 적는다 (OpenAI). strict 도구 사용(Anthropic). 옛 Gemini SDK는 2025-11-30에 지원 종료. 성공 기준과 시험을 첫 프롬프트보다 앞에 둔다(Anthropic)
성공 기준 ·평가 원칙	1·2·8	Anthropic ⁴⁹ · OpenAI ⁵⁰ · Husain-Shankar ⁵¹	OpenAI 호스팅 Evals는 2026-10-31 읽기 전용, 11-30 종료 예정 — 2026-10-05에 출발하면 4주에 읽기 전용, 9주 첫날 종료(세트를 어디에 둘지 고를 때). 기준은 구체적·측정 가능하게(Anthropic). 점수보다 이진 판정(Husain-Shankar)
오류 분석 ·세트	4~26	Husain-Shankar ⁵¹ · Anthropic 에이전트	다양한 기록 100개, 처음 30개는 직접 라벨 (Husain-Shankar). 실제 실패에서 20~50개,

범주	쓰는 주	선택지	고를 때 쓰는 사실
키우기		평가 ⁵² · OWASP 에이전트 보안 9월 ¹⁸	일어나야 할 때와 일어나면 안 될 때를 둘 다, 100%인 평가는 회귀만 잡는다(Anthropic). 관찰된 주입·메모리 오염·도구 오용 실패마다 회귀 시험, 시험 고정물에 비밀값·실고객 데이터 금지(OWASP)
통계	2·7·10·21·23	Miller ³³ · Bowyer 외 ³⁴ · Hanley·Lippman·Hand ³⁶ · OpenIntro ⁵³ · NIST 7.2.4 ³⁵	같은 문항의 짝 차이(Miller). 수백 문항 미만에서 CLT 오차 막대는 불확실성을 크게 과소추정 → Wilson(Bowyer 외·NIST). 실패 0건 ≈ 3/n(Hanley·Lippman·Hand). 비율의 구간과 표본 크기(OpenIntro 6장)
웹 입구	5	FastAPI(+ MDN HTTP 개요) ⁵⁴ · Flask ⁸ — 경제 표시: http.server ⁹	FastAPI는 1.0 이전·PyPI 릴리스 320개, Flask는 1.0(2018) 이후·64개. Flask 배포 문서는 개발 서버를 운영에 쓰지 말라고, http.server 문서는 운영에 권하지 않는다고 적는다
검색 원리·평가	검색 가지	IR 교과서 8장 ⁵⁵ · BEIR ⁵⁶ · Contextual Retrieval ¹⁴ · RAG 원 논문 ⁵⁷	BM25는 "robust baseline"(BEIR). 200,000토큰보다 작으면 통째로 넣는 기준선(Contextual Retrieval). RAG 원문은 미세조정 레시피다. 시험 모음·정밀도·재현율(IR 교과서)
벡터 저장소	검색 S3	pgvector ⁵⁸ · Chroma ⁵⁹ · Qdrant ²⁴	프로세스 안(Chroma 클라이언트·Qdrant 로컬 모드) ↔ 별도 서버(Postgres·Docker). Qdrant는 기본값이 암호화·인증 없음이라는 경고. Chroma의 마지막 릴리스 2026-05-05(2026-10-05까지 153일)
행동 설계	행동 가지	Building effective agents ⁶⁰ · ReAct ⁶¹ · MCP 명세 ⁶²	워크플로와 에이전트를 나누고 가장 단순한 해법부터 찾으라고 적는다. 추론과 행동을 번갈아 남기는 루프의 원형(ReAct). MCP는 판 번호가 날짜, 개정 간격 평균 157.5일
에이전트 SDK·사람 승인	행동 A2~	OpenAI Agents SDK ⁶³ · Claude Agent SDK ²⁰ · LangGraph ²¹	승인 지점: <code>needs_approval</code> (OpenAI) · <code>deny</code> 규칙과 <code>canUseTool</code> (Claude — 자동 승인된 도구에는 콜백이 불리지 않는다) · <code>interrupt</code> (LangGraph — 재개하면 그 노드를 처음부터 다시 돌아 앞쪽 부수 효과는 먹들이어야 한다). LangGraph 1.0 "no breaking changes until 2.0". OpenAI Agents SDK를 엮은 출발 코호트의 6개월 안 연성 단절 100%(13개) ²⁸
보안 위험 목록	5·가지	OWASP LLM Top 10(2025·2026판) ⁶⁴ · Willison ²² · Greshake 외 ²³ · Beurer-Kellner 외 ⁶⁵	2026판에서 Excessive Agency 6위 → 3위. 세 다리(Willison). 모델이 읽는 문서 속 숨은 지시(Greshake 외). 탐지가 아니라 구조로 막는 설계와 그 비용(Beurer-Kellner 외)

범주	쓰는 주	선택지	고를 때 쓰는 사실
보안 완료 조건	1·5·9·가지·25	OWASP 비밀값 관리 ¹⁶ · 에이전트 보안 ¹⁸ · 로그 ¹⁷ · OpenAI 사람 승인 ¹³	「보안 완료 조건」 표
책(골라 읽기)	전체	Huyen ⁶⁶ · Shankar·Husain ⁶⁷ · Kohavi 외 ⁶⁸	Huyen은 평가 장이 RAG·에이전트 장보다 앞이다. Shankar·Husain은 2026-10-31 출간 예정 — 출간 전에는 두 저자의 무료 자료로 대신한다. Kohavi 외는 운영 중 대조 실험 — 25주 그림자 실행의 다음 단계

주

- 1 일자리 문장을 쓰지 않는 근거 — Brendon Bernard, 2025-07-30, 「Experience Requirements Have Tightened Amid the Tech Hiring Freeze」, Indeed Hiring Lab, <https://www.hiringlab.org/2025/07/30/experience-requirements-have-tightened-amid-the-tech-hiring-freeze/> — 미국 기술직 공고(경력 명시분)에서 5년 이상 요구가 2022년 2분기 37%에서 2025년 2분기 42%로 올랐다. Sophie Magnet, 2025-04-04(갱신 2026-04-23), 「AI Engineer Job Outlook 2026 [Research on 1,000 Job Postings]」, 365 Data Science, <https://365datascience.com/career-advice/career-guides/ai-engineer-job-outlook-2025/> — "Only 2.5% of positions target junior professionals with 0-2 years of experience"(Glassdoor 미국 공고 903건). 앞의 것은 기술직 전체, 뒤의 것은 강의 판매 회사의 분석이다. ↩
- 2 저자 측정 기록 data/calls.jsonl(측정 자료 <https://dosi.dev/books/ai-engineer-six-months/data/>) — 그 앱(과업 1)의 유효 호출에서 입력·출력 토큰의 중앙값을 따로 썼다. 사전 등록된 측정량이 아니다. ↩
- 3 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), 규정 원문 data/sets/policy_true.txt(sha256 ad450e68b214b036...) · 규칙 엔진 data/m1_policy.py(측정 자료 <https://dosi.dev/books/ai-engineer-six-months/data/>). 누리벌항공은 측정용 가상 설정이고 실존 회사와 무관하다. 1주 형식표의 금액은 이 엔진으로 계산했다. ↩
- 4 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.4. 채점자는 grok-4.7-gpt-6-astra:gpt-5.6-sol 셋이고, 정답을 아는 쌍에서 gpt-6-astra가 놓친 3쌍이 모두 개인 사정 질문이었다. ↩
- 5 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.3 — 1차 모델 claude-haiku-4-5, 도전 세트 240개의 첫 실행. 하위 유형별 건수(개인 사정 14개 중 13개, 금액 단정 20 · 형식만 9, 형식 실패 10건 중 9건)는 사전 등록하지 않은 사후 기술이다. ↩
- 6 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.1 — 시연 t1d-03(세이버, 출발 57일 전, 60일 전 구매)을 다섯 번 돌린 다섯 번째 실행, 원출력 data/calls.jsonl seq 33. 같은 입력이 첫 실행부터 네 번째 실행까지는 맞았다. ↩
- 7 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.4 — 입력 50개(시연 10 + 사전 지정 40)를 다섯 번씩 물었고, 24.0%의 입력 부트스트랩 95% 구간은 [12.0%, 36.0%]다. 반복 실행은 실패율에 넣지 않았다. ↩
- 8 Pallets, 「Deploying to Production」(Flask 3.1.x 문서), <https://flask.palletsprojects.com/en/stable/deploying/> · 「Quickstart」, <https://flask.palletsprojects.com/en/stable/quickstart/> — 3.1.3(2026-02-19), PyPI 릴리스 64개, 열람 2026-10-05. "Do not use the development server when deploying to production. It is intended for use only during local development." ↩
- 9 Python Software Foundation, 「http.server — HTTP servers」(Last updated 2026-10-05, Python 3.14.8 문서), <https://docs.python.org/3/library/http.server.html> — 열람 2026-10-05. "http.server is not recommended for production. It only implements basic security checks." ↩
- 10 Google, 2026-07-21, 「Gemini API release notes」, <https://ai.google.dev/gemini-api/docs/changelog> — "The sampling parameters temperature, top_p and top_k are now deprecated." Anthropic, 「Model deprecations」(날짜 없는 문서, 2026-10-05 열람), <https://platform.claude.com/docs/en/about-claude/model-deprecations>, API parameter deprecations 표 — "Returns a 400 error when set to a non-default value on Claude 4.7 and later models and Claude Mythos Preview." 같은 절은 API가 폐기한 파라미터를 다루는 ↩

방식이 모델에 따라 다르다고 적고("depends on the model"), Python SDK v1.0 이상(열람일 판 1.11.0 포함)은 세 파라미터를 없애서 넘기면 TypeError가 난다고 적는다.

- 11 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.6 · 루브릭 data/m2_rubric_v2.txt. v2는 v1 결과를 본 뒤 썼고, 동점률은 열린 쌍 100쌍의 값이다. v2 결과는 사전 등록 판정값이 아니라 사후 기술이다. ↩
- 12 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 3.3 — 채점자 셋(grok-4.7gpt-6-astra-gpt-5.6-sol), 정답을 아는 쌍 85 · 열린 쌍 100. 두 숫자는 차례로 재채점 뒤집힘과 루브릭(기준 문장) 뒤집힘이다. ↩
- 13 OpenAI, 「Human-in-the-loop」 (OpenAI Agents SDK 문서), https://openai.github.io/openai-agents-python/human_in_the_loop/ — v0.8.0(2026-02-05)에 들어온 기능, SDK 0.23.1, 열람 2026-10-05. "RunState.from_json() and RunState.from_string() do not authenticate the snapshot or the person submitting it." · "Callable approval rules fail closed when the SDK cannot safely inspect the arguments." ↩
- 14 Anthropic, 2024-09-19(수정 2026-01-07), 「Introducing Contextual Retrieval」, <https://www.anthropic.com/engineering/contextual-retrieval> — 열람 2026-10-05. 지식 기반이 200,000토큰(약 500쪽)보다 작으면 RAG 없이 전부 프롬프트에 넣을 수 있다는 기준선을 적는다. "This method can reduce the number of failed retrievals by 49% and, when combined with reranking, by 67%." 수치는 당시 모델-임베딩 기준이다. ↩
- 15 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.2·§ 2.3. 하위 유형으로 나는 건수는 사후 기술이다. 비교 모델 gpt-5.5의 과업 1 실패는 개인 사정 질문의 금액 단정 6건이 전부였다. ↩
- 16 OWASP Cheat Sheet Series, 「Secrets Management Cheat Sheet」 (마지막 커밋 2026-10-04), https://cheatsheetseries.owasp.org/cheatsheets/Secrets_Management_Cheat_Sheet.html — 열람 2026-10-05. 8.1 "Consider enabling secrets detection at the developer level to avoid checking secrets into code before commit/PR either in the IDE, as part of test-driven development, or via pre-commit hook." 9.2의 순서는 Revocation → Rotation → Deletion이다. ↩
- 17 OWASP Cheat Sheet Series, 「Logging Cheat Sheet」 (마지막 커밋 2026-10-04), https://cheatsheetseries.owasp.org/cheatsheets/Logging_Cheat_Sheet.html — 열람 2026-10-05, 「Data to exclude」 절. "The following should usually not be recorded directly in the logs, but instead should be removed, masked, sanitized, hashed, or encrypted"(뒤에 접근 토큰·개인 식별 정보·암호 키 등이 이어진다). ↩
- 18 OWASP Cheat Sheet Series, 「AI Agent Security Cheat Sheet」 (첫 커밋 2025-12-17, 마지막 2026-10-04), https://cheatsheetseries.owasp.org/cheatsheets/AI_Agent_Security_Cheat_Sheet.html — 열람 2026-10-05. 1절 "Implement per-tool permission scoping (read-only vs. write, specific resources)." 9절 "High-impact actions cannot execute without a valid, unexpired, parameter-bound approval" · "do not store secrets or live customer data in test fixtures." 2026-09-30~10-04 닷새에 여섯 번 고쳐진 문서라 인용에 열람일을 붙인다. ↩
- 19 Scott Chacon·Ben Straub, 2014, 『Pro Git』 2판, Apress, <https://git-scm.com/book/en/v2> — 온라인판 무료(CC BY-NC-SA 3.0), 빌드 2.1.450(2026-05-25), 열람 2026-10-05. 1~3장(Getting Started·Git Basics·Git Branching). ↩
- 20 Anthropic, 「Agent SDK overview」 · 「Permissions」, <https://code.claude.com/docs/en/agent-sdk/overview> · <https://code.claude.com/docs/en/agent-sdk/permissions> — Python claude-agent-sdk 0.2.163, TypeScript 0.3.289, 열람 2026-10-05. "If a deny rule matches, the tool is blocked, even in bypassPermissions mode." user-input 페이지: "The callback never fires for auto-approved tools." ↩

- 21 LangChain Inc., 「Interrupts」(LangGraph 문서), <https://docs.langchain.com/oss/python/langgraph/interrupts> · 1.0 발표 글(2025-10-22), <https://www.langchain.com/blog/langchain-langgraph-1dot0> — langgraph 1.2.12(2026-09-21), 열람 2026-10-05. "no breaking changes until 2.0" · "Side effects called before interrupt must be idempotent". ↩
- 22 Simon Willison, 2025-06-16, 「The lethal trifecta for AI agents: private data, untrusted content, and external communication」, <https://simonwillison.net/2025/Jun/16/the-lethal-trifecta/> · 연재 <https://simonwillison.net/series/prompt-injection/> — 열람 2026-10-05. 세 다리: "Access to your private data" · "Exposure to untrusted content" · "The ability to externally communicate". ↩
- 23 Kai Greshake 외, 2023-02-23(v1)·2023-05-05(v2), 「Not what you've signed up for: Compromising Real-World LLM-Integrated Applications with Indirect Prompt Injection」, AISec '23, <https://arxiv.org/abs/2302.12173> — 열람 2026-10-05. 사용자가 아니라 모델이 읽는 문서·웹 페이지에 심긴 지시를 다룬다. ↩
- 24 Qdrant, 「Quickstart」, <https://qdrant.tech/documentation/quickstart/> · qdrant-client README, <https://github.com/qdrant/qdrant-client> — 서버 v1.19.1, 열람 2026-10-05. "By default, Qdrant starts with no encryption or authentication" · "Local mode is useful for development, prototyping and testing." ↩
- 25 Hamsa Bastani·Osbert Bastani·Alp Sungu·Haosen Ge·Özge Kabakcı ·Rei Mariman, 2025-06-25, 「Generative AI without guardrails can harm learning: Evidence from high school mathematics」, PNAS 122(26):e2422633122, <https://doi.org/10.1073/pnas.2422633122> — "students actually perform worse than those who never had access (17% reduction in grades for GPT Base)". 2023년 가을 학기, 9~11학년 약 1,000명, GPT-4 기반이다. "What is the answer?"는 같은 논문의 기계 분석에 나온다. ↩
- 26 Judy Hanwen Shen·Alex Tamkin, 2026-01-28(v1)·2026-02-01(v2), 「How AI Impacts Skill Formation」, arXiv:2601.20245, <https://arxiv.org/abs/2601.20245> — "using AI assistance to complete tasks that involve this new library resulted in a reduction in the evaluation score by 17% or two grade points (Cohen's $d = 0.738$, $p = 0.010$)". 라이브러리는 Trio, 참가자는 크라우드워커이고, 퀴즈는 과제 직후에 잤다. 평균 50% 대 67%는 두 저자의 글 「How AI assistance impacts the formation of coding skills」(Anthropic, 2026-01-29, <https://www.anthropic.com/research/AI-assistance-coding-skills>)에 있다. 이 17%는 퀴즈 백본율의 차이라서 Bastani의 17%(대조군 평균 대비)와 기준이 다르다. ↩
- 27 같은 논문 § 5(그림 9) — "The control group (No AI) reported higher self-reported learning (on a 7-point scale)". 자기보고 차이는 탐색적 분석이다. ↩
- 28 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m3-stack-results.md 「결과 먼저」·부록 § 2, 사전 등록 data/prereg-m3.md § 1·§ 7 — 출발 달마다 공급자 SDK·모델(과 프레임워크·평가 도구)을 고정된 출발 코호트의 6개월 안 연성 단절 비율. 판정량은 프레임워크와 평가 도구를 없지 않은 세 핵심 스택 하한의 최소값 39.3%이고, 스택별로는 OpenAI 45.0%, Anthropic 52.0~76.0%, Google 39.3~57.1%다. 낮은 문구의 범위 39.3%~100.0%는 사전 등록대로 핵심 스택 하한의 최소값(Google의 하한)에서 전 조합 상한의 최댓값(OpenAI Agents SDK를 얻은 조합, 13개 코호트)까지다. 얻은 조합까지 모두 넣은 범위는 29.2%~100.0%이고, 아래 끝은 평가 도구 Inspect만 얻은 OpenAI 조합(24개 코호트)이다. 평가 도구만 얻은 조합은 promptfoo 60.0~92.9%, Inspect 29.2~75.0%(24개 코호트)다. 핵심 스택의 단절은 대부분 공식 SDK의 메이저 교체였고, API 계열에서 온 단절은 0이었다. ↩
- 29 OpenAI, 「Deprecations」(2026-10-05 열람), <https://developers.openai.com/api/docs/deprecations> — "Generally available models: At least 6 months." Anthropic, 「Model deprecations」(2026-10-05 열람), <https://platform.claude.com/docs/en/about-claude/model-deprecations> — "providing at least 60 days' notice before model retirement for publicly released models". Google, 「Gemini deprecations」(Last updated 2026-10-01 UTC, 2026-10-05 열람), <https://ai.google.dev/gemini-api/docs/deprecations> — ↩

"The shutdown dates listed in the table indicate the earliest possible dates on which a model might be retired."

- 30 같은 세 페이지(2026-10-05 열람). OpenAI는 gpt-4-1106-preview의 종료일을 2026-03-26과 2026-10-23으로, code-davinci-002의 종료일을 2023-03-23과 2024-01-04로 함께 적는다. Google은 embedding-001의 출시일을 2024-04-09로 적는데 changelog는 2023-12-13이다. 오기는 몇 행뿐이지만, 지원 상태를 적을 때 열람일을 붙이는 이유다. ↩
- 31 저자 측정(사전 등록 2026-10-05 10:32, 한국 시각 — 결과보다 먼저), data/m1-m2-results.md § 2.5 § 2.3 — 1차 모델 claude-haiku-4-5, 비교 모델 gpt-5.5, 같은 도전 세트 240개의 첫 실행. 두 모델이 함께 틀린 5건은 모두 가측상 증빙으로 수수료 면제를 묻는 질문이었다(사후 확인). ↩
- 32 저자 측정(사전 등록 2026-10-05 10:43, 한국 시각 — 계수보다 먼저), data/a2-results.md — 한 사람의 시스템에서 모델-엔진-SDK가 바뀐 사건 14건(검열 없는 사건): 사건 14일 뒤 사건 이전 검사 본문의 중앙 97%가 공백 정규화 뒤 글자 그대로 남았지만, 남은 검사가 못 잡고 실사용에서 드러난 실패가 2건 확인됐다. 남았다는 것은 본문이 그대로라는 뜻이지 계속 맞게 작동했다는 뜻이 아니다. ↩
- 33 Evan Miller, 2024-11-01, 「Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations」, arXiv:2411.00640, <https://arxiv.org/abs/2411.00640> — 열람 2026-10-05. "paired differences represent a 'free' reduction in estimator variance". ↩
- 34 Sam Bowyer-Laurence Aitchison-Desi R. Ivanova, 2025-03-03(v1)·2025-05-28(v3), 「Position: Don't Use the CLT in LLM Evals With Fewer Than a Few Hundred Datapoints」, ICML 2025, <https://arxiv.org/abs/2503.01747> — 열람 2026-10-05. 수백 문항 미만에서 CLT 오차 막대는 "usually dramatically underestimating uncertainty". ↩
- 35 NIST/SEMATECH, 「e-Handbook of Statistical Methods」 7.2.4.1·7.2.4.2(핸드북 첫 화면 Last updated 2022-04-27), <https://www.itl.nist.gov/div898/handbook/prc/section2/prc241.htm> · <https://www.itl.nist.gov/div898/handbook/prc/section2/prc242.htm> — 열람 2026-10-05. Wilson 구간은 "recommended by Brown, Cai and DasGupta (2001) and Agresti and Coull (1998)"이고, 이유로 "the lower limit cannot be negative"를 든다. ↩
- 36 James A. Hanley-Abby Lippman-Hand, 1983-04-01, 「If Nothing Goes Wrong, Is Everything All Right? Interpreting Zero Numerators」, JAMA 249(13):1743–1745, doi:10.1001/jama.1983.03330370053031, <https://pubmed.ncbi.nlm.nih.gov/6827763/> — 실패 0건 n회의 95% 상한 $\approx 3/n$. 계산 예: $n = 50$ 이면 정확한 상한 5.82%, $3/n = 6\%$. ↩
- 37 Python Software Foundation, 「The Python Tutorial」, <https://docs.python.org/3/tutorial/index.html> — "Python 3.14.8 documentation", 열람 2026-10-05. 3.15.0 정식이 2026-10-09 예정이라 이 주소의 판이 곧 바뀐다. 3~5장, 7.2.2, 8장. ↩
- 38 Promptfoo, Inc., 「Intro」, <https://www.promptfoo.dev/docs/intro/> — 0.123.1(2026-09-18), MIT, 열람 2026-10-05. "The goal: test-driven LLM development, not trial-and-error." 2026-03-09 인수 발표 글: "Promptfoo will remain open source and we will continue to serve users and customers." ↩
- 39 UK AI Security Institute-Meridian Labs, 「Inspect」, <https://inspect.aisi.org.uk/> — inspect_ai 0.3.276(2026-10-02), MIT, 열람 2026-10-05. 평가 하나 = Task(Dataset + Solver + Scorer). ↩
- 40 Astral, 「uv」 문서, <https://docs.astral.sh/uv/> — 0.12.23(2026-10-03), 1.0 이전, 열람 2026-10-05. "An extremely fast Python package and project manager, written in Rust." 잠금 파일은 uv.lock 이다. ↩
- 41 PyPA, 「Installing packages using pip and virtual environments」, <https://packaging.python.org/en/latest/guides/installing-using-pip-and-virtual-environments/> · 「Tool recommendations」, <https://packaging.p> ↩

- [python.org/en/latest/guides/tool-recommendations/](https://docs.python.org/3/library/venv.html) · venv 문서 <https://docs.python.org/3/library/venv.html> — pip 26.2.1, 열람 2026-10-05. 안내서는 "purposefully does not make a blanket recommendation"이라고 적는다. `pip lock` 설명은 "EXPERIMENTAL"로 시작한다.
- 42 Poetry 프로젝트, 「Basic usage」, <https://python-poetry.org/docs/basic-usage/> — 2.5.1(2026-09-20), 열람 2026-10-05. "Even if you develop alone, in six months when reinstalling the project you can feel confident the dependencies installed are still working even if your dependencies released many new versions since then."
- 43 MIT(강사 Anish·Jon·Jose), 2026-01-12~01-23, 「The Missing Semester of Your CS Education」 2026판, <https://missing.csail.mit.edu/> — CC BY-NC-SA, 열람 2026-10-05. 아홉 강 중 일곱째 강(2026-01-21)이 "Agentic Coding"이다.
- 44 OpenAI, 「Text generation」, <https://developers.openai.com/api/docs/guides/text> — 갱신일 표기 없음, Python SDK openai 3.24.0, 열람 2026-10-05. "Pinning your production applications to specific model snapshots" · "Building tests and evaluation suites that measure prompt behavior..."
- 45 OpenAI, 「Structured Outputs」, <https://developers.openai.com/api/docs/guides/structured-outputs> · 「Function calling」, <https://developers.openai.com/api/docs/guides/function-calling> — 열람 2026-10-05. "While both ensure valid JSON is produced, only Structured Outputs ensure schema adherence."
- 46 Anthropic, 「Prompt engineering overview」, <https://platform.claude.com/docs/en/build-with-claude/prompt-engineering/overview> — 열람 2026-10-05. 전체 셋: "A clear definition of the success criteria for your use case" · "Some ways to empirically test against those criteria" · "A first draft prompt you want to improve".
- 47 Anthropic, 「Tool use」, <https://platform.claude.com/docs/en/agents-and-tools/tool-use/overview> · 「Prompt caching」, <https://platform.claude.com/docs/en/build-with-claude/prompt-caching> — Python SDK anthropic 1.11.0, 열람 2026-10-05. 도구 정의의 `strict: true`.
- 48 Google, Gemini API 문서(Last updated 2026-09-23 UTC), <https://ai.google.dev/gemini-api/docs> · <https://ai.google.dev/gemini-api/docs/structured-output> · <https://ai.google.dev/gemini-api/docs/function-calling> — SDK google-genai 2.28.0, 열람 2026-10-05. 옛 SDK google-generativeai의 README: "All support for this repository ended permanently on November 30, 2025."
- 49 Anthropic, 「Define success criteria and build evaluations」, <https://platform.claude.com/docs/en/test-and-evaluate/develop-tests> — 열람 2026-10-05. 기준은 Specific·Measurable·Achievable·Relevant, 설계 원칙은 "Be task-specific" · "Automate when possible" · "Prioritize volume over quality".
- 50 OpenAI, 「Evaluation best practices」, <https://developers.openai.com/api/docs/guides/evaluation-best-practices> · 「Working with evals」, <https://developers.openai.com/api/docs/guides/evals> · 「Deprecations」, <https://developers.openai.com/api/docs/deprecations> — 열람 2026-10-05. 두 안내서도 같은 두 날짜를 적는다. 다음 인용과 공지일·이전 안내서 제목은 Deprecations의 「2026-06-03: Evals platform」 절에 있다. "Existing evals become read-only"(2026-10-31) · "The Evals dashboard and API are scheduled to shut down"(2026-11-30). 폐기 공지는 2026-06-03이고, 이전 안내서 제목은 "Moving from OpenAI Evals to Promptfoo"다.
- 51 Hamel Husain·Shreya Shankar, 「AI Evals: Everything You Need to Know」 (페이지 표기 게시 2026-09-18 · 수정 2026-09-21, 같은 주소의 최초 보관 2025-06-23), <https://hamel.dev/blog/posts/evals-faq/> · Hamel Husain, 2024-03-29, 「Your AI Product Needs Evals」, <https://hamel.dev/blog/posts/evals/> · Shreya Shankar 외, 2024, 「Who Validates the Validators?」, UIST '24, <https://arxiv.org/abs/2404.12272>

- 열람 2026-10-05. "Start with 100 diverse traces and annotate at least the first 30 yourself." · "Binary evaluations force clearer thinking and more consistent labeling."
- 52 Mikaela Grace·Jeremy Hadfield·Rodrigo Olivares·Jiri De Jonghe, Anthropic, 2026-01-09(수정 2026-03-18), 「Demystifying evals for AI agents」, <https://www.anthropic.com/engineering/demystifying-evals-for-ai-agents> — 열람 2026-10-05. "20-50 simple tasks drawn from real failures is a great start" · "An eval at 100% tracks regressions but provides no signal for improvement." pass`k는 k번 모두 성공한 비율이다.
- 53 David Diez·Mine Çetinkaya·Rundel·Christopher D Barr, 2019, 『OpenIntro Statistics』 4판, OpenIntro, <https://www.openintro.org/book/os/> — PDF 무료(CC BY-SA 3.0), 열람 2026-10-05. 6장 Inference for Categorical Data.
- 54 FastAPI, 「Tutorial – User Guide」, <https://fastapi.tiangolo.com/tutorial/> — 0.142.2(2026-09-30), PyPI 릴리스 320개. MDN, 「Overview of HTTP」(최종 수정 2026-08-21), <https://developer.mozilla.org/en-US/docs/Web/HTTP/Guides/Overview> — 둘 다 열람 2026-10-05.
- 55 Christopher D. Manning·Prabhakar Raghavan·Hinrich Schütze, 2008, 『Introduction to Information Retrieval』, Cambridge University Press, 8장 「Evaluation in information retrieval」, <https://nlp.stanford.edu/IR-book/html/htmledition/evaluation-in-information-retrieval-1.html> — 온라인판 무료, 열람 2026-10-05.
- 56 Nandan Thakur 외, 2021-04-17(v1)·2021-10-21(v4), 「BEIR: A Heterogenous Benchmark for Zero-shot Evaluation of Information Retrieval Models」, NeurIPS 2021 Datasets and Benchmarks Track, <https://arxiv.org/abs/2104.08663> — 열람 2026-10-05. "BM25 is a robust baseline". 원제의 철자 "Heterogenous"는 그대로다.
- 57 Patrick Lewis 외, 2020-05-22(v1)·2021-04-12(v4), 「Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks」, NeurIPS 2020, <https://arxiv.org/abs/2005.11401> — 열람 2026-10-05. 원문은 사전학습 생성 모델과 문서 색인을 묶는 "general-purpose fine-tuning recipe"다.
- 58 pgvector 프로젝트, README, <https://github.com/pgvector/pgvector> — 0.8.7(2026-10-01), "supports Postgres 13+", 열람 2026-10-05.
- 59 Chroma, 「Introduction」 · 「Clients」, <https://docs.trychroma.com/docs/overview/introduction> · <https://docs.trychroma.com/docs/run-chroma/clients> — chromadb 1.5.9(2026-05-05), Apache-2.0, 열람 2026-10-05. 메모리 안 `chromadb.Client()`, 디스크 `PersistentClient`, 별도 서버(Docker).
- 60 Erik S·Barry Zhang, Anthropic, 2024-12-19(수정 2026-08-10), 「Building effective agents」, <https://www.anthropic.com/engineering/building-effective-agents> — 열람 2026-10-05. "finding the simplest solution possible, and only increasing complexity when needed". 워크플로는 "orchestrated through predefined code paths"다.
- 61 Shunyu Yao 외, 2022-10-06(v1)·2023-03-10(v3), 「ReAct: Synergizing Reasoning and Acting in Language Models」, ICLR 2023, <https://arxiv.org/abs/2210.03629> — 열람 2026-10-05.
- 62 Model Context Protocol, 「Specification」 2026-07-28, <https://modelcontextprotocol.io/specification/2026-07-28> — 열람 2026-10-05. 사양 판 2024-11-05·2025-03-26·2025-06-18·2025-11-25·2026-07-28, 간격 141·84·160·245일(평균 157.5일, 계산). Anthropic이 2024-11-25에 공개했고 2025-12-09에 Linux Foundation 산하 Agentic AI Foundation에 기여했다.
- 63 OpenAI, 「OpenAI Agents SDK」, <https://openai.github.io/openai-agents-python/> — 0.23.1(2026-10-02), 열람 2026-10-05.

- 64 OWASP GenAI Security Project, 「OWASP Top 10 for LLM Applications」 2025판(2024-11-17 게시), <https://genai.owasp.org/llm-top-10/> · 2026판(자료 페이지 2026-08-03, 발표 2026-09-01), <https://genai.owasp.org/resource/owasp-genai-llm-top-10-2026/> — 열람 2026-10-05. 2026판 03이 Excessive Agency다(2025판 6위). ↩
- 65 Luca Beurer-Kellner 외, 2025-06-10(v1)·2025-06-27(v3), 「Design Patterns for Securing LLM Agents against Prompt Injections」, <https://arxiv.org/abs/2506.08837> — 열람 2026-10-05. "principled design patterns for building AI agents with provable resistance to prompt injection". ↩
- 66 Chip Huyen, 2025, 「AI Engineering」, O'Reilly Media, <https://www.oreilly.com/library/view/ai-engineering/9781098166298/> · 한국어판 「AI 엔지니어링」, 한빛미디어, 2025-09-30, https://www.hanbit.co.kr/store/books/look.php?p_code=B3535685426 — 열람 2026-10-05. 3·4장(평가)이 5장(프롬프트)·6장(RAG와 에이전트)보다 앞선다. ↩
- 67 Shreya Shankar·Hamel Husain, 「Evals for AI Engineers」, O'Reilly Media, 출간 예정 2026-10-31, <https://books.google.com/books?vid=ISBN9798341660724> — 카탈로그 열람 2026-10-05. ↩
- 68 Ron Kohavi·Diane Tang·Ya Xu, 2020-04-02, 「Trustworthy Online Controlled Experiments」, Cambridge University Press, <https://books.google.com/books?vid=ISBN9781108724265> — 카탈로그 열람 2026-10-05. ↩

출처와 등급

본문이 기댄 출처의 무게, 저자가 켤 값의 모델·판본·날짜, 독자가 공개 자료로 다시 계산할 수 있는 것을 적는다. 고른 것은 장마다 무게를 받친 출처이고, 전체 서지와 원어 인용은 각 장의 각주에 있다.

등급과 표기

표기	뜻
[F]	1차 문서를 직접 열어 확인했다. 그 문서가 적은 범위 안에서다
[전송]	1차 문서를 찾지 못했거나 열지 못해, 그것을 옮긴 글로만 확인했다
[가설]	확인된 사실을 이어 붙인 저자의 추론이다
저자 측정	저자가 켤 값. M1(그 앱)·M2(채점자)·M3(스택 추적)·A2(납는 검사)는 결과보다 먼저 등록했고, M3x(모델 ID 생존)는 등록 이전의 탐색이다
저자 기록	저자의 운영 기록. 공개 저장소 llmux·soma·work의 커밋·PR은 누구나 연다(공개). 교정 원장과 운영 기록은 비공개이고 원장은 집계 A1만 싣는다(비공개)

본문과 각주에는 이 표기를 붙이지 않았다. 유보는 본문이나 각주에 문장으로 적었고, 대괄호 표기는 이 부록에만 있다. 공급자 문서와 공식 표는 2026-10-05에 연 판본이다.

장별 주요 출처

「돌아간다」

앵커	등급
llmux 첫 공개 커밋 e499c23(2026-06-13)과 같은 날의 v0.1.0, 그 커밋의 연구 노트 ("Dogfooding on 2026-06-13" — 자기 보고)	저자 기록 (공개)
공개 이력 이전(출발점 KarpelesLab/teamclaude)과 공개 main 집계(사람의 승인 0)	저자 기록 (공개)

앵커	등급
OpenAI, 2023-06-13 함수 호출(신뢰성 수치 없음) · swyx, 2023-06-30(2013년과의 비교는 겐 값이 아니다)	[F]
Anthropic, 2024-05-30 도구 사용 · OpenAI, 2024-08-06 Structured Outputs · Anthropic, 2024-11-25 MCP	[F]
Anthropic, 2025-02-24 Claude Code(자기 보고) · OpenAI, 2025-05-16 Codex · Anthropic, 2025-05-22 Claude 4	[F]
누리별항공 환불 앱과 측정 전에 고정한 시연 열 개	저자 측정 (M1)

「열 번 중 열 번」

앵커	등급
사전 예측 15%, 시연 10/10, 240개의 층 가중 실패율 17.1%, 출력 분산, 과업 2와 비교 모델 gpt-5.5	저자 측정 (M1)
llmux 커밋 c883c72(2026-06-14) — 데모 GIF를 "the old faked clip"이라 부르며 바꿨다	저자 기록 (공개)
Hanley-Lippman-Hand, 1983(JAMA) · ICH E1, 1994 · FDA, 2004-11 라벨 · 2005-02-28 판매 중단 권고 · 2006 정보 페이지 · Yousry 외, 2006(분모가 세 질환의 합이라 비교는 어렵)	[F]
Klarna, 2024-02-27 · Form F-1/A, 2025-09-02 — 회사 자체 측정	[F]

「채점자를 잔다」

앵커	등급
Weyuker, 1980(표지 낱씨의 끝자리가 잘렸다) · Barr 외, 2015 서베이	[F]
'시험 오라클'의 첫 용례가 Howden(1978)이라는 것 — 서베이를 거쳤다	[전승]
Cleverdon, 1962-10 보고서 · 1991 회고	[F]
1953년 시험(ASTIA-Uniterm)에서 두 팀이 '관련'에 합의하지 못해 공동 결과를 내지 못한 일 — Cleverdon의 두 글을 거쳤다. 그가 일반에 볼 수 있는 유일한 기록이라 한 Gull의 논문은 열지 않았다	[전승]
Swanson, 1965-01 — Sparck Jones(1981)가 옮긴 문장을 따랐다	[전승]
Voorhees, 2000-2007	[F]

앵커	등급
Cleverdon, 1970-10 — 새 판정으로 다시 매긴 순위도, 견준 원래 Cranfield II 순위도 문서 200건·질문 42개 부분 모음의 것	[F]
Zheng 외, 2023 · Dubois 외, 2024와 Li 외, 2024(둘 다 모델 순위 수준의 상관)	[F]
같은 240개·같은 규칙 엔진으로 본 두 모델 · 185쌍의 채점자 측정 · 열린 쌍의 채점자별 선택 수(사후 기술)	저자 측정 (M1·M2)
Mata v. Avianca — 진술서(ECF 32-1, 날짜 없는 화면 발췌) · 심리 속기(ECF 52) · 제재 의견서(ECF 54)	[F]

「고친 자리」

앵커	등급
교정 원장 214행(2026-07-06~10-02, 서로 다른 교정 204건)과 규칙·기록 스크립트 ·2026-07-05 자체 감사 — 행은 의역, status 칸은 기록한 쪽의 자기 신고	저자 기록(비공개, 원장은 집계 A1)
llmux PR #80·#81·#83·#85(2026-07-13~14) · soma-work PR #226(2026-09-17)	저자 기록(공개)
개방·축 코딩 — Gérrouville-Farrell, 2025-06-21 · Husain-Shankar FAQ 2025-07-01판 · Corbin-Strauss, 1990 · Stol 외, 2016	[F]

「삼십 퍼센트」

앵커	등급
Bastani 외, 2025-06-25(PNAS) — 한 학교·한 학기, 시험으로 켜 단기 학습	[F]
Murphy·Winkler, 1977 · Brier, 1950	[F]
Bjork, 1994 · Koriat·Bjork, 2005	[F]
Bahrick(1979)의 실험(Bjork의 요약)	[전승]
Hogarth·Lejarraga·Soyer, 2015 · Hogarth, 2003(동료 심사 없는 에세이)	[F]
Hogarth, 2001 책과 장티푸스 의사 일화(Lewis Thomas, 1983) — 2015년 논문과 본문 검색 조각으로만 봤다	[전승]
Kahneman·Klein, 2009	[F]
Shen·Tamkin, 2026 · Perry 외, 2023	[F]

앵커	등급
LLM 제품을 짓는 일에 친절/사악 틀을 직접 실증한 연구가 없다는 판단(Brooker, 2019, 2026-10-05 검색 범위)	[가설]
그 열 번과 17.1% · 원장의 분모 감사	저자 측정(M1) · 저자 기록(A1)

「채점표가 거짓말할 때」

앵커	등급
OpenAI, 2024-08-13 · 2026-02-23(SWE-bench Verified) · METR, 2026-06-26(GPT-5.6 Sol)	[F]
Thakur 외, 2021(BEIR) · Shankar 외, 2024(참가자 9명)	[F]
Goodhart, 2021 원고 · Strathern, 1997 · Campbell, 1976(쪽수는 2011년 재수록본)	[F]
Meta, 2025-04-05 · Al-Dahle · LMArena · Singh 외, 2025(50점 상승은 시뮬레이션)	[F]
Anthropic "Model IDs and versioning" · Google release notes 2025-06-26 · Cherny, 2026 · Bischof, 2025 · OpenAI Evals 폐기 공지	[F]
결과를 본 뒤 고친 채점 기준(사후 기술) · 같은 240개의 모델 교체 · 변경 사건 14건	저자 측정(M2·M1·A2)
누락 커밋 llmux 31e3800-969f324, soma-work ec7bedf · soma-work PR #253·#265·#267·#268	저자 기록(공개)
원장 211행 · 발동 집계와 생존 검사 판정(2026-09-30~10-02, 26개 가운데 24개 제거)	저자 기록(비공개)

「여섯 달」

앵커	등급
OpenAI, 2023-03-01-03-14 스냅샷 지원 약속 · 2023-06-13(2023-07-20 갱신)	[F]
OpenAI "Deprecations" · Anthropic "Model deprecations" — Anthropic 표의 권장 대체 칸은 공지 당시의 기록이 아니다	[F]
스택의 6개월 안 연성 단절 39.3%~100.0%	저자 측정(M3)

앵커	등급
모델 ID의 KM 중앙 생존 499일, 6개월 창의 노출 비율	저자 측정 (M3x, 탐색)
openai-python — README 2023-05-18판과 커밋 #441(2023-05-09) · v1.0.0 릴리스 노트(CHANGELOG.md는 1.1.2부터)·Discussion #631·#742 · 1.0.0의 PyPI 업로드 기록	[F]
Dev Bootcamp·Codeup 홈페이지 스냅샷 · LinkedIn Jobs on the Rise 2025·2026	[F]
Dev Bootcamp의 폐업 이메일(2017-07-12)과 18주 과정 — EdSurge 보도를 거쳤다 · Codeup의 중단 이메일(2023-12-28 정오) — San Antonio Report(2024-01-02)를 거쳤다	[전승]
Shankar 외, 2024(기준이 출력을 본 뒤에 굳는다는 것은 실무자 9명의 질적 연구에서 끌어낸 함의)	[F]
같은 240개의 17.1%와 2.5% · llmux 변경 사건 여섯	저자 측정 (M1·A2)

「마지막 줄」(결론)

앵커	등급
확인된 것(시연과 240개, 두 종류의 쌍, 남은 검사와 누락, 스택 비율)과 204건의 원장	저자 측정(M1·M2·A2·M3) · 저자 기록(A1)
Bastani 외, 2025 · Shen·Tamkin, 2026 · Klarna, 2024·2025 · METR 시간 지평(2026-05-08 갱신)과 원자료	[F]
친절/사악 틀의 직접 실증이 없다는 판단	[가설]
Moffatt v. Air Canada, 2024-02-14(소액 사건이라 선례 구속력은 약하다) · Mata v. Avianca 진술서·Tr. 29(질문한 날이 적혀 있지 않다)	[F]

본문과 각주에서 [전승]·유보를 밝힌 자리

- 「채점자를 켜다」 — '시험 오라클'의 첫 등장("서베이는 ... 적고") [전승] · 1953년 시험(각주: "Cleverdon의 기술에 기댄다") [전승] · Swanson의 비판("옳겨 적은 대로라면") [전승]
- 「삼십 퍼센트」 — Bahrick의 실험("그가 요약한") [전승] · Hogarth 2001과 장티푸스 의사 일화(각주: "같은 논문이 인용한") [전승] · 직접 실증의 부재(검색 날짜와 범위)

[가설]

- 「여섯 달」 — Dev Bootcamp의 폐업 이메일(각주: 보도를 거쳐 옮겼다) [전승] · Codeup의 중단 이메일(각주: 같은 보도가 적은 시각) [전승]
- 「마지막 줄」 — "검색한 범위에서 찾지 못했다" [가설]

판본 고정

측정	모델·판본	날짜(한국 시각)
M1	앱 claude-haiku-4-5-20251001, 비교 모델 gpt-5.5	2026-10-05 10:38~10:52
M2	채점자 grok-4.7(응답의 model 값 grok-4.7-build) · gpt-6-astra · gpt-5.6-sol. 등록된 claude-sonnet-4-6은 시도 10번 모두 502여서 등록해 둔 교체 경로로 바꿨다	첫 기준 10:59~11:35(sol은 11:36부터), 고친 기준 11:50~12:01
M1·M2 호출	로컬 게이트웨이 llmuxe를 거친 Anthropic Messages 형식(anthropic-version 2023-06-01), max_tokens 4096, temperature는 공급자 기본값. SDK 없이 Python 표준 라이브러리(3.14.6에서 확인). 원장 3,773줄(무효 10), 모델 대체 0	연결 확인 10:34
M3	출발 코호트 40개(2023-01-01~2026-04-01 매달 1일), 창 182일. 단절로 세는 SDK 판은 등록 규칙(data/prereg-m3.md § 6.1·§ 6.2) 대로 메이저 줄의 첫 판 가운데 단절을 명시한 것이다 — openai 0.26.0-1.0.0-2.0.0-3.0.0, anthropic 0.3.0-0.28.0-0.39.0-1.0.0, google-generativeai 0.8.0, google-genai 0.3.0-0.4.0-0.7.0-0.8.0-1.0.0. google-genai 2.0.0은 노트가 주 사용 경로(GenerateContent)에는 영향이 없다고 적어 세지 않았다(판단 J1)	관측 종료 2026-10-05
M3x	공급자 공식 기록 258행(텍스트 모델 ID 157개). 공식 changelog와 릴리스 노트가 호환 단절로 선언한 SDK 판 18개 — openai-python 1.0.0-1.3.0-2.0.0-3.0.0, anthropic-sdk-python 0.28.0-0.39.0-1.0.0, google-genai 11개. M3의 주 판정은 이 목록을 쓰지 않고, 민감도 S2에서만 단절로 더한다	관측 종료 2026-10-05
A2	llmuxe9c2bb6 · soma-work 0477d8f1(둘 다 2026-10-02의 main) · 비공개 저장소 하나, 사건 뒤 14일 창	2026-07-06~10-02(A1과 같은 기간)

측정 재현

파일은 모두 책 웹판의 data/ 폴더(<https://dosi.dev/books/ai-engineer-six-months/data/>)에 있고, 명령은 그 폴더에서 돌린다. 목록은 data/README.md다. 등록 시각은 등록 문서를 저자의 비공개 저장소에 처음 커밋한 시각이고, 세 문서는 머리말에 커밋한 뒤에만 실행하거나 센다고 적었다 — data/prereg.md(M1·M2) 2026-10-05 10:32, 결과 data/m1-m2-results.md 첫 커밋 12:09 · data/prereg-m3.md(M3) 10:32, 결과 data/m3-stack-results.md 11:18 · data/prereg-a2.md(A2) 10:43, 결과 data/a2-results.md 10:54.

- M1·M2 — 고정값 data/prereg_values.py · data/prompts.py · data/m2_rubric_v2.txt(결과를 본 뒤 쓴 기준) · data/sets/. 코드 data/m1_*.py 일곱 (m1_policy.py가 규칙 엔진) · data/m2_pairs.py · data/m2_run.py · data/m_data.py · data/gw.py · data/grade.py · data/stats.py · data/analyze.py · data/test_m1.py. 원출력 data/calls.jsonl, 재계산 출력 data/results.md · data/results.json. `python3 m1_gen.py --check` · `python3 -m unittest` · `python3 analyze.py` 가 LLM 호출 없이 세트와 숫자를 다시 만든다. 다시 재는 명령(`python3 m1_run.py check` 부터 `python3 m2_run.py grade v2 --workers 4` 까지)과 실제로 돌린 순서는 data/README-measurements.md와 data/m1-m2-results.md §6에 있다. Messages 형식의 게이트웨이(M_LLMUX_URL)와 빈 원장(M_LEDGER)이 필요하고, 다시 잰 값은 이 기록과 다를 수 있다.
- M2 사후 기술 — data/m2-family-posthoc.md · data/m2_family_posthoc.py(`python3 m2_family_posthoc.py`): 열린 쌍 100개의 채점자별 선택 수와 Wilson 구간. 결과를 본 뒤 더한 기술 통계다.
- M3 — 자료 data/m3-stack-components.csv(807행) · data/m3-surfaces.csv, 코드 data/m3_stack.py, 판독 스크립트 data/m3-gen/stack/. `python3 m3_stack.py` · `python3 m3_stack.py --selftest` (합성 시험 21개).
- M3x — data/m3-summary.md · data/m3_stats.py(`python3 m3_stats.py`) · data/m3-gen/m3x/. data/m3-corrections.md에는 2026-10-05에 공식 문서를 다시 대조해 고친 12자리와, 다시 계산해 바뀐 숫자의 전·후가 있다. 고친 뒤에도 M3 판정량의 계산 출력은 바이트 단위로 같았다.

- M3 보존 — 공급자 페이지·변경 기록·패키지 색인의 스냅샷은 저작권 때문에 실지 않고, data/m3-preserve-MANIFEST.md가 압축본 둘과 파일마다의 sha256으로 고정한다: m3x-snapshots.tar.gz
d7b2c40cfede0cbc47a6e0f0d5e2ccc06ca8a347ad100ee37307933ced0e44fb ,
m3stack-raw.tar.gz
6cc545926a1892c0067c27f372055bbdfb2c5a90c4276f1ff22d00e4b1e87339 .
- A1 — 교정 원장은 비공개다. 공개하는 것은 집계 스크립트 data/a1_ledger_stats.py · data/a1_audit.py와 결과 data/a1-audit-results.md뿐이다(집계 명령은 그 첫머리). 방법은 읽을 수 있어도 원장에서 다시 셀 수는 없다.
- A2 — 계수 코드 data/a2_count.py · data/a2_checks.py · data/a2_tables.py, 명령은 data/a2-results.md 「재현」. 사건 목록은 data/prereg-a2.md §2에, 사건별 표는 data/a2-results.md에 있다. 세 저장소 가운데 하나가 비공개라 사건 파일(a2-events.json)과 원자료(a2-raw.csv · a2-summary.json · 누락 후보와 검토 표)는 실지 않았다. 「재현」이 부르는 보조 스크립트 셋(a2_selftest.py · a2_raw_compact.py · a2_miss_review.py)도 실지 않았다. 같은 수를 독자가 다시 낼 수는 없다.

오류는 원본보다 깔끔한 쪽으로

모델을 판정자로 세워 모델들의 순위를 매기는 Arena-Hard의 논문(Li 외, arXiv:2406.11939)은 2024-10-14의 v2 초록에, 프롬프트 500개로 모델 하나를 20달러에 채점하는 이 벤치마크가 "98.6% correlation with human preference rankings"를 이룬다고 적었다. 같은 판의 표 1은 상위 20개 모델 전체의 순위를 사람 선호 순위와 맞췄고, 거기서 Spearman 상관은 93.2%다. 98.6%는 표 3의 값이고, 문체를 보정하고 어려운 프롬프트로 비교한 가장 좋은 조건에서 나왔다. 2024-06-17의 v1 초록은 같은 자리에 "89.1% agreement"와 25달러를 적었다. 초록만 옮기는 인용에는 표 1의 93.2%도, 98.6%가 나온 조건도, 판마다 숫자와 그 숫자를 부르는 말이 달랐다는 사실도 따라가지 않는다. 숫자는 남이 옮기기 전에, 같은 논문의 요약에서 이미 원본보다 깔끔했다. 「채점자를 켜다」는 98.6%와 93.2%를 함께 적고, 98.6%가 어느 조건의 값인지 밝혔다.

오답이 없는 연습
AI 엔지니어의 여섯 달

Zhuge Hyuk
2026