

差し戻す権限は、確かめられるときだけ役に立つか

AIのチームの「意見で直させる」と「動かして確かめて直させる」を、2×2で比べる

Does Loop-Back Authority Pay Only When It Can Verify? Separating Opinion- and Verifier-Triggered Revision in a Production Agent Team

湯村 信司 (Shinji Yumura) AQUA合同会社 2026年10月16日

この報告が確かめた論文

B. Agachan, M. van Duijn, A. Zohrehvand (ライデン大学) 「Loop-Back Authority in LLM Agent Teams」 arXiv:2609.14767 (2026年10月)

元の論文が推測にとどめ、試していなかった「上司役は、確かめられるときだけ役に立つ」を、本番の開発チームで、2×2の実験で実際に試した。

3.6 倍

意見だけで差し戻す上司を置いたときの料金（100題あたり \$1.65 → \$5.96）。時間も 37秒 → 65秒（中央値）

差なし

出来（審査の合格・5つの観点の点）は、4つの形のどの比べでも、偶然の範囲を超えて変わらなかった

10%

上司の1回目の指摘のうち、動かせば確かめられるもの（不具合・頼まれた機能の欠け）。残りの9割は見た目・書き方・頼まれていない拡張の提案

ABSTRACT

Agachan et al. (2026) found that a manager agent with loop-back authority made multi-agent reports slightly worse, and conjectured, without testing it, that such authority “earns its cost when it can verify something and becomes a liability when it can only opine.” We test this conjecture directly in the five-agent team that builds React apps in YUA's Dev Mode, by crossing two independently switchable revision triggers in a 2×2 paired design: an opinion trigger (an LLM reviewer that may send files back on its own judgment, up to two rounds) and a verifier trigger (the app is executed in a browser preview and runtime errors are sent back, up to two rounds). All four arms ran concurrently on 24 tasks × 2 runs (192 generations), judged blind by two LLM judges on five 1–5 dimensions (Krippendorff's α =

0.93 on the overall score) and by a 0/1/2 pass judge. No contrast reached significance for quality: opinion authority changed the overall score by +0.08 (95% CI -0.09 to +0.25) and the pass rate by +4.2 points, while verifier authority changed them by +0.06 and +6.2 points. Opinion authority, however, multiplied cost by 3.6× (\$1.65 → \$5.96 per 100 tasks), added 29 s per task, and grew the code by 86 lines (+27%). As in the original paper, first drafts were indistinguishable from the flat arm, and revision inflated them: lines +21% and defensive constructs per line +42% (an analogue of the paper's +53% hedging). Only 10% of the reviewer's first-round critiques concerned runtime-checkable defects; 90% were stylistic, aesthetic, or unrequested extensions ($\kappa = 0.91$ between two classifiers). The verifier fired in only 3 of 48 runs, all of which ended working and passing, at near-zero cost. A blinded human comparison of 24 A-B pairs, in which the rater used both apps, also found no reliable preference (11 vs. 8, 5 ties; $p = 0.31$). We find no quality penalty from opinion authority, contrary to the original result, but confirm its mechanism—revision adds material without adding verified value—and show that its main price is paid in money, time, and code size rather than in judged quality.

要旨

Agachan ほか（2026）は、部下に差し戻して直させる権限を持つ上司役の AI がいると、AI のチームの報告書がわずかに悪くなることを示した。そのうえで「上司は、何かを確かめられるときだけ役に立ち、意見しか言えないときは足を引っ張る」と述べたが、これは推測で、試してはいない。

本稿は、この推測を、YUA の開発モードで React のアプリを作る5人のチームで直接確かめた。差し戻しのきっかけを2つに分け、それぞれを入れたり切ったりできるようにした。1つは**意見の差し戻し**で、見直し役の AI が自分の判断でファイルを差し戻す（最大2回）。もう1つは**確かめの差し戻し**で、アプリを実際に動かし、出たエラーを渡して直させる（最大2回）。この2つを掛け合わせた4つの形を、同じお題（24題×2回）で同じ時刻に動かし、192回の結果を、作り方を知らせない2つの審査役（5つの観点で1～5点、 $\alpha = 0.93$ ）と合否の審査で採点した。

出来は、どの比でも偶然の範囲を超えて変わらなかった。意見の差し戻しは総合点を +0.08（95%区間 -0.09～+0.25）、合格を +4.2ポイント変え、確かめの差し戻しは +0.06 と +6.2ポイントだった。一方、意見の差し戻しは料金を3.6倍（100題あたり \$1.65 → \$5.96）にし、1題あたり29秒長くし、コードを86行（+27%）増やした。元の論文と同じく、最初の版は差し戻しのない形と見分けがつかず、直しで行数が21%、守りの書き方（1行あたり）が42%増えた（元の論文の「ぼかし語」+53%に当たる）。見直し役の1回目の指摘のうち、動かせば確かめられるもの（不具合・頼まれた機能の欠け）は10%で、残りは見た目・書き方・頼まれていな

い拡張の提案だった（2つの振り分けの一致 $\kappa = 0.91$ ）。確かめの差し戻しは48回中3回しか起きず、3回とも動いて合格で終わり、料金はほとんど増えなかった。どちらの形かを隠して2つのアプリを人が実際にさわって比べた24組でも、はっきりした差はなかった（意見の差し戻しが良い11組・なしが良い8組・同じ5組、 $p = 0.31$ ）。

元の論文とちがい、意見の差し戻しで出来が下がることは見られなかった。しかし「確かめの直しは、中身を増やすが、確かな価値は増やさない」という仕組みは同じで、その代わりは、出来ではなく、お金・時間・コードの量で払われていた。

1 はじめに

AIのチームでは、上司役のAIが部下役の仕事を見直し、足りないと思えば差し戻して直させる形がよく使われる。人の組織と同じように、見直しが入るほど良くなると考えられがちである。しかし、AIの上司の見直しは、本当に仕事を良くしているのだろうか。

2026年10月、Agachan ほか [1] は、この問いに答える実験を発表した。同じ5つの役のチームを、上司が差し戻せる形と、差し戻せない対等な形で動かし、43製品の報告書を比べたところ、差し戻せる形のほうが「役立ち」と「文章の明快さ」がわずかに低かった。差し戻されるたびに、文章には「～かもしれない」といったほかしの言葉が増えていた。著者らはここから、**上司が役に立つのは、何かを確かめられるときだけで、意見しか言えないときは足を引っ張る**と結論した。ただし、この区別は推測で、実験では試されていない。著者ら自身も、将来の課題の最初に「確かめ役が差し戻すのか、上司が判断で差し戻すのかを操作して比べる」ことを挙げている。

本稿は、その比べを行う。YUAの開発モードには、ちょうど2種類の差し戻しがある。見直し役のAIが**意見で**差し戻すものと、作ったアプリをプレビューで**動かして確かめ**、エラーが出たら差し戻すものである。この2つを独立に入れたり切ったりして、2×2の4つの形を同じ条件で比べた。問いは3つある。

1. 意見で差し戻す権限は、出来を下げるか（元の論文の結果は、コードを作る仕事でも起きるか）
2. 確かめで差し戻す権限は、出来を上げるか。2つを合わせると、どうなるか
3. 出来のほかに、差し戻しは何を変えるか（料金・時間・コードの中身）

2 元の論文

Agachan ほか [1] の設定を、本稿と比べられる範囲で書く。

- **チーム**：調べ役・分析役・書き役・批評役・上司の5役。5つのモデル（GPT-5.4・Gemini-3.1-Pro など）を役ごとにくじで割り当てる

- **比べる形**：対等な形（上司は意見を言うが差し戻せない）と、差し戻せる形（上司が毎回の仕事を見て差し戻せる。最大2回）
- **仕事**：ゲーミングノートPC 43製品の分析報告書（仕様と口コミのまとめ）
- **採点**：5つのモデルの審査役が、明快さ（構成・一貫性・簡潔さ）と役立ち（行動につながるか・原因の分析・戦略の深さ）を1～5点で採点。対応のある t 検定など
- **結果**：対等な形のほうが、役立ち（ $d = 0.42$ 、 $p = 0.009$ ）と明快さ（ $d = 0.34$ 、 $p = 0.030$ ）が高かった。差し戻せる形は、ぼかしの言葉が53%多く（1,000語あたり 5.03 と 3.30）、トークンを51.5%多く使い、34.3%遅かった。差し戻し1回ごとに明快さが 0.14点下がった。最初の版は対等な形と見分けがつかず、差は直して生まれていた
- **審査の一致**：Krippendorff の α は明快さで 0.08～0.17、役立ちで 0.22～0.30 と低い

報告書は、正しさを確かめられる部分（仕様）と、確かめられない部分（分析や文章）が混ざった仕事である。コードを作る仕事は、動かせば確かめられる部分が大きい。著者らの推測が正しければ、確かめで差し戻す権限は役に立ち、意見で差し戻す権限は役に立たない（または悪くする）はずである。

3 実験のしかた

3.1 チーム

YUA の開発モードのチームは5人である。まとめ役（Yua）が作るファイルを決め、3人の作り手（Noa：CSS と設定、Toa：型と部品、Kaito：画面の React）が分担して書き、見直し役（Ren）がすべてのファイルを見直す。作るモデルは全員 Gemini 3.5 Flash-Lite で、本番と同じである。本番では、Ren の見直しに加えて、作ったアプリをプレビューで動かし、エラーが出たら自動で直させる仕組みも動いている。

3.2 4つの形

2つの差し戻しを、それぞれ入れる・切るの2通りにした。差し戻しの回数の上限は、どちらも2回にそろえた（元の論文と同じ）。

	確かめの差し戻しなし	確かめの差し戻しあり
意見の差し戻しなし	A：どちらもなし（最初の版がそのまま仕上がり）	C：動かして、エラーが出たときだけ直させる
意見の差し戻しあり	B：Ren の意見で直させる（元の論文の「差し戻せる形」に当たる）	D：両方（本番に近い）

表1 2×2の4つの形。A と B の比べが、元の論文の比べ（対等な形と差し戻せる形）に当たる。

- **意見の差し戻し**：Ren が全ファイルを読み、ファイルごとに点・問題・直し方を書く。Ren は自分でアプリを動かすことはできない。本番の Ren は、点のほかに CSS とクラス名の決まった確かめも使うが、この実験では外し、AI の意見だけで決めさせた。
- **確かめの差し戻し**：できたアプリをブラウザのプレビューで動かし、エラーが出たら、そのエラーを渡して「動くように直して」と頼む。エラーが出なければ何もしない。
- **差し戻しがないとき**：B・D 以外では Ren を呼ばない（A・C の Ren は、AI を呼ばずに合格を返す）。

3.3 上司の厳しさの校正

本番の Ren は甘い。合格の線は75点で、前回の測定 [2] では96回中6回しか差し戻していなかった。下見で合格の線を95点に上げて、12回中1回だった。これでは差し戻しがほとんど起きず、意見で差し戻す権限の効き目を測れない。元の論文の上司は、決まった確かめの手引きに沿って、43製品中32製品で差し戻していた。そこで本実験では、元の論文の上司に合わせて、**1回目の見直しでは、改善点をファイルごとに具体的に挙げさせ、挙げたファイルは差し戻す**ことにした。2回目以降は、95点に届かなければ差し戻す。この設定は本番とちがう。これは「意見で直させる上司」を実際に働かせるための操作であり、結果を読むときはこの点に注意が要る（10章）。

3.4 お題と測り方

- **お題**：日本語の小さなアプリ24題（電卓・やることリスト・カンバン・スネークゲーム・グラフ・3D の太陽系・住宅ローンの計算など）を、各形で2回ずつ。1つの形につき48回、合わせて192回
- **同じ条件**：4つの形を4台のサーバーで同じ時刻に動かした（2026年10月15日 23:55～16日 1:00）。AI の混み具合による差をそろえるためである
- **動いたか**：画面が出た・エラーが出ない・真っ白でない・最初のボタンを押してもエラーが出ない、の4つすべて
- **合否の審査**：Gemini 3.8 Flash が、画面の写真とコードを見て0～2点を3回つけ、真ん中の点を使う。2点で合格（前回までの報告と同じ）
- **5つの観点の審査**：元の論文の採点に合わせ、Gemini 3.8 Flash と Gemini 3.7 Flash がそれぞれ、頼みの実現・見た目・使いやすさ・コードの明快さ・コードの簡潔さを1～5点でつけた。先に気づいたことを書かせてから点をつけさせた（元の論文と同じ手順）。どちらの審査役にも、どの形で作ったかは知らせていない。前の3つを「製品の点」、後の2つを「コードの点」、5つの平均を「総合点」とし、2人の平均を使う
- **コードの中身**：返事の記録から、ファイルごとの最初の版と最後の版を取り出し、行数と「守りの書き方」の数を比べた。守りの書き方は、元の論文の「ほかしの言葉」に当たるもので、`try`・`?.`・`??`・`if (!x) return`・`typeof x === 'undefined'`・`Array.isArray`・`|| []`などの、想定外の値に備える書き方を決めた一覧で数えた
- **料金と時間**：AI を呼んだ回数とトークンを役ごとに記録した。料金は Gemini 3.5 Flash-Lite の値段（100万トークンあたり入力 \$0.30、出力と考えた分 \$2.50）で計算した

- 統計：**単位は（お題, 回）の48組で、4つの形の結果を対応させて比べる。合否は McNemar の正確検定、点是对应のある t 検定と Wilcoxon 検定。2×2 の主効果（その差し戻しがある2つの形と、ない2つの形の差の平均）と交互作用は、48組ごとの差で求め、95%区間はお題ごとにまとめて4,000回取り直して求めた。お題をランダム効果にした混合モデルでも確かめた。主な12の比べには Holm の補正をかけた

A の1回（dashboard の1回目）は、AI への接続が途中で切れて返事が空になったため、同じ形でやり直した。ほかに、このような失敗はなかった。

4 出来は変わったか

指標	A なし	B 意見	C 確かめ	D 両方
動いた	47/48	47/48	48/48	48/48
審査で合格（2点）	77.1%	81.3%	83.3%	87.5%
総合点（1～5）	4.64	4.69	4.67	4.78
製品の点	4.57	4.68	4.59	4.77
コードの点	4.75	4.72	4.79	4.78
頼みの実現	4.66	4.84	4.75	4.91
見た目	4.53	4.59	4.42	4.64
使いやすさ	4.53	4.59	4.61	4.78
コードの明快さ	4.77	4.88	4.82	4.92
コードの簡潔さ	4.72	4.56	4.75	4.65

表2 4つの形の出来（各48回）。点は2人の審査役の平均。

合格も総合点も、A から D に向かって少しずつ高い。しかし、差はどれも偶然の範囲を超えなかった（表3）。元の論文と同じ比べである A と B では、合格は 7組で A だけ、9組で B だけが合格し（ $p = 0.80$ ）、総合点の差は $+0.05$ （ $d_z = 0.06$ 、 $p = 0.68$ ）だった。

効果	合格	総合点	製品の点	コードの点
意見の差し戻し（主効果）	+4.2pt -6.2～+13.5	+0.08 -0.09～+0.25	+0.14 -0.04～+0.33	-0.02 -0.20～+0.15
確かめの差し戻し（主効果）	+6.2pt -3.1～+15.6	+0.06 -0.10～+0.20	+0.06 -0.14～+0.25	+0.05 -0.05～+0.16
交互作用	±0.0pt -14.6～+16.7	+0.05 -0.20～+0.31	+0.08 -0.24～+0.39	+0.02 -0.20～+0.23

表3 2×2の効果と、その95%区間（お題ごとに取り直し）。どれも0をまたぐ。お題をランダム効果にした混合モデルでも、総合点の意見の効果は+0.05（ $p=0.61$ ）、確かめは+0.03（ $p=0.78$ ）、交互作用は+0.05（ $p=0.71$ ）。Holmの補正後の p は、主な12の比ですべて1.0。

観点ごとに見ると、意見の差し戻しは、製品の点（+0.14）とコードの明快さ（+0.10）を上げ、**コードの簡潔さを下げる**（-0.13）方向にあった。簡潔さが下がる向きは、AとB（-0.16、Wilcoxon $p=0.098$ ）でも、CとD（-0.10）でも同じだった。元の論文でも、差が最もはっきりした観点の1つが簡潔さ（ $d=0.44$ ）だった。ただし本実験では、どれも偶然の範囲である。

まとめると、**意見で差し戻す権限は、出来を下げなかった**。元の論文の主な結果（差し戻せる形が悪くなる）は、ここでは再現しなかった。ただし区間の下の端は-0.09で、元の論文の差（明快さ・役立ちとも約-0.09）と同じくらいの小さな悪化までは、否定できない。

5 何が変わったか：料金・時間・コードの量

出来が変わらなかった一方で、意見の差し戻しは、ほかのものをはっきり変えた。

■ 意見の差し戻しなし (A・C) ■ あり (B・D)

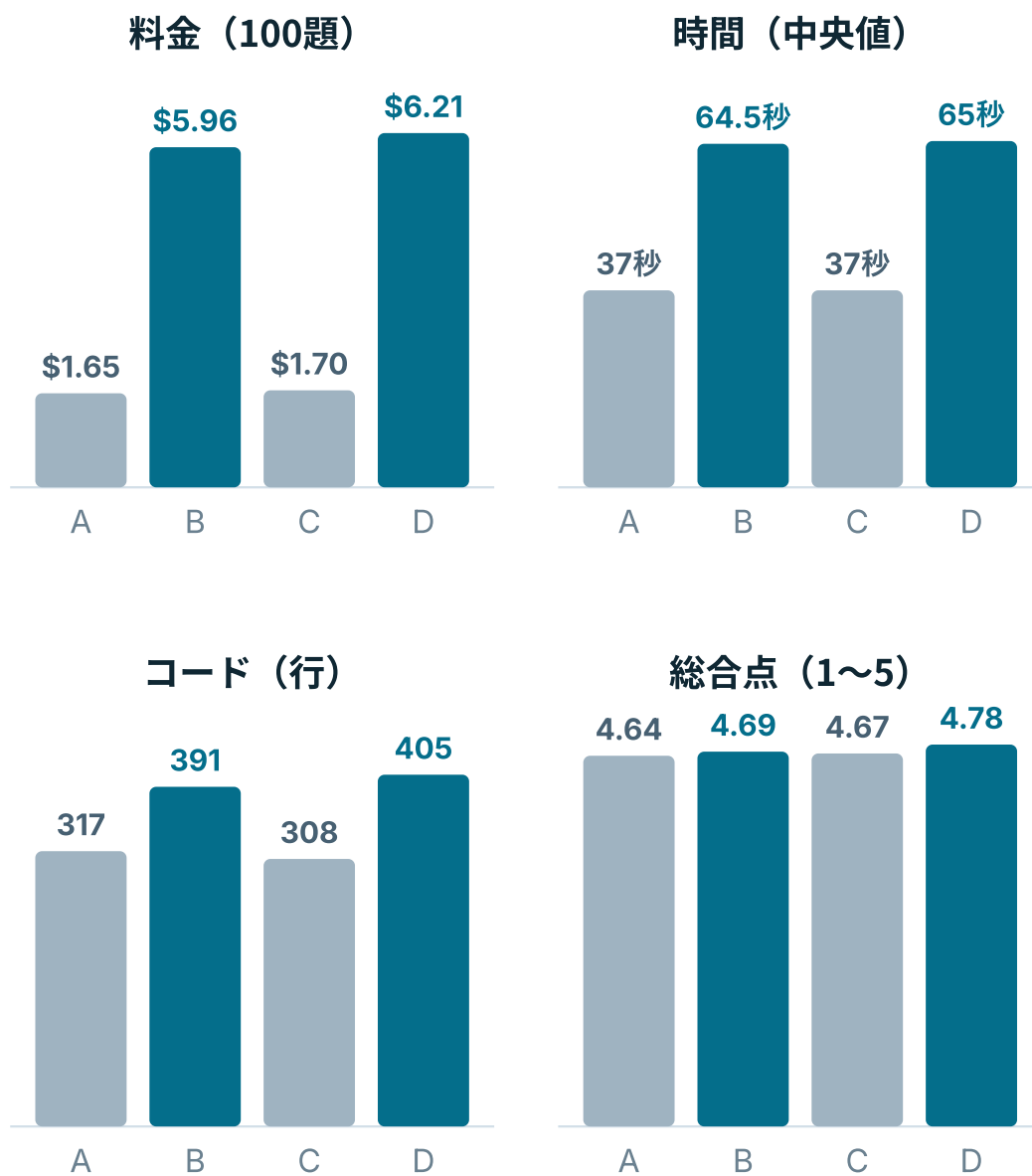


図1 4つの形の、料金 (100題あたり)・時間 (中央値)・最後のコードの行数・総合点。意見の差し戻しは、料金を3.6倍、時間を1.7倍、コードを1.27倍にしたが、総合点はほとんど動かなかった。

指標	A	B	C	D	意見の効果 (95%区間)
差し戻し (Ren)	0	73回	0	73回	1回 23組・2回 25組
AI を呼んだ回数	4.8	11.3	4.9	11.3	2.3倍
トークン (1題)	13,707	56,313	13,912	58,399	4.1倍
料金 (100題)	\$1.65	\$5.96	\$1.70	\$6.21	+\$4.4 (+4.0~+4.8)
時間 (中央値)	37秒	64.5秒	37秒	65秒	+28.7秒 (+23.3~+33.4)
最後のコード (行)	317	391	308	405	+85.6行 (+63.9~+108.2)

表4 差し戻しにかかったもの。意見の効果は、B・DとA・Cの差の平均（いずれも $p < 0.001$ ）。確かめの差し戻しは、どれもほとんど変えなかった（料金 +\$0.2・時間 +0.5秒・行数 +2.0、いずれも $p > 0.4$ ）。

5.1 最初の版は同じで、直して膨らむ

元の論文は、差し戻せる形の最初の版が対等な形の報告書と見分けがつかず、差は直して生まれたことを示した。本実験でも同じだった。Bの最初の版（差し戻される前）と、差し戻しのないAの仕上がりは、行数（324と317行、 $p = 0.57$ ）も、守りの書き方の密度（100行あたり0.41と0.50、 $p = 0.23$ ）も変わらなかった。

B（意見の差し戻し）の中で	最初の版	最後の版	変化	$d_z \cdot p$
行数	324	391	+21%	1.15・<0.001
守りの書き方（100行あたり）	0.41	0.59	+42%	0.45・0.003
最初の版から残った割合		82%		

表5 直しによるコードの変化（Bの48回）。Dでも守りの書き方は0.52→0.66（+28%、 $d_z = 0.46$ 、 $p = 0.003$ ）と増えた。元の論文では、直して文章が38語増え、ほかしの言葉が53%増え、最初の版の88%が残った。

直しのあと、コードは21%長くなり、想定外の値に備える書き方は、1行あたりで42%増えた。元の論文で、直しのあとに「～かもしれない」が53%増えたことに、よく似ている。言われたことに応えようとして、作り手は確かめられない心配に備える書き方を足していく。ただし、守りの書き方の密度を形どうしで比べると（AとBの最後の版）、差は偶然の範囲だった（+0.09、 $p = 0.20$ ）。最初の版どうしの偶然の差が、それを打ち消していたためである。はっきり言えるのは、同じ回の中で、直しの前と後を比べたときの増え方である。

5.2 差し戻しの回数と出来

元の論文では、差し戻し1回ごとに明快さが0.14点下がった。本実験では、そのような下がり方は見られなかった。Bの中で、差し戻しの回数（1回か2回）と総合点の関係はほぼなく（ $p=0.07$ 、 $p=0.64$ ）、Dではむしろ、2回差し戻された回のほうが点が高い向きだった（合否の審査の傾き $+0.19$ 、 $p=0.036$ ）。2回目の差し戻しは「95点に届かない」ときに起きるので、回数は上司の判断で決まっている。そのため、この関係は原因と結果ではない。

6 上司は何を直させたか

BとDでRenが書いた指摘は、合わせて910件あった。1回目の見直しでは、1回につき平均8.4件（B）と8.9件（D）を挙げた。これを6つの種類に振り分けた。振り分けはGemini 3.8 Flashが行い、Gemini 3.7 Flashにも独立に同じことをさせた。2つの一致は93.4%（ $\kappa=0.91$ ）、「動かせば確かめられるか」の2つに分けたときは95.9%（ $\kappa=0.81$ ）だった。

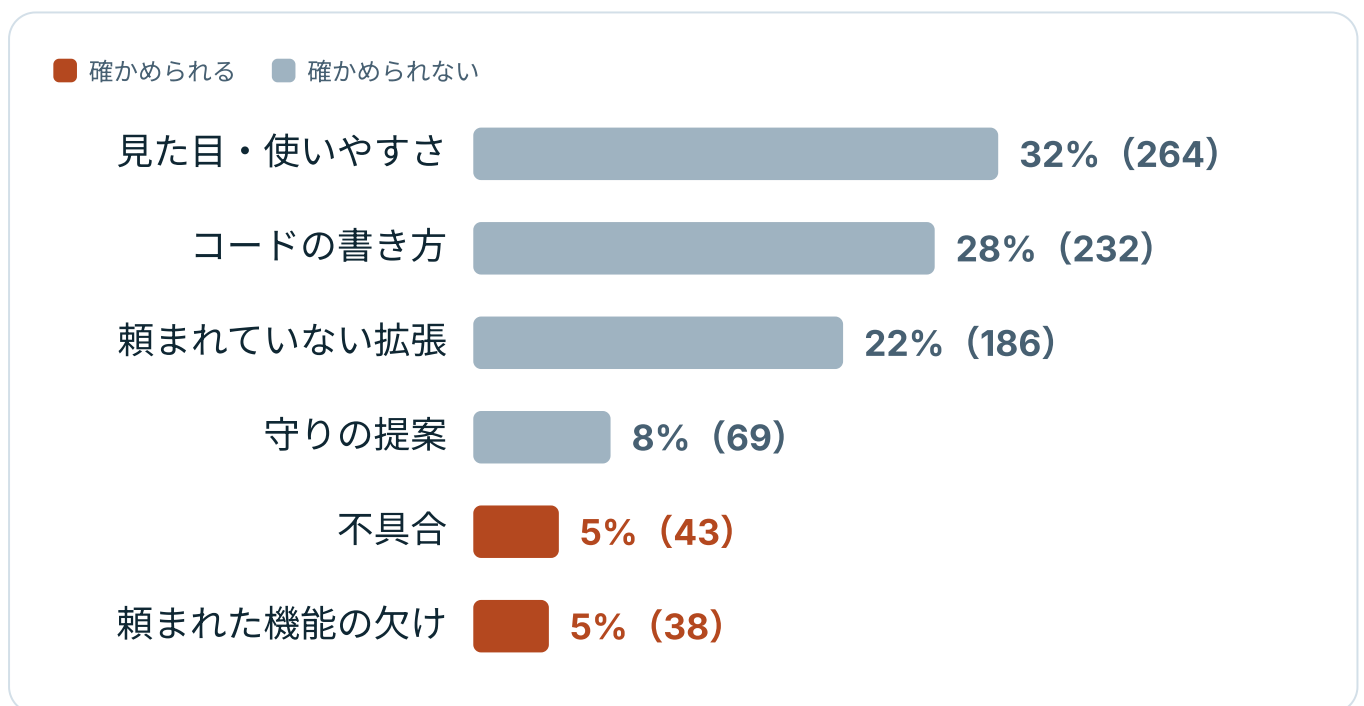


図2 Renの1回目の指摘（832件）の種類。動かせば確かめられるもの（不具合・頼まれた機能の欠け）は合わせて10%だった。Gemini 3.7 Flashの振り分けでも11%（不具合6%・欠け5%）。

- **見た目・使いやすさ（32%）**：「スマホ表示のメディアクエリが不足」「フォーカスの状態や高コントラストへの対応」など
- **コードの書き方（28%）**：「型定義に readonly が考慮されていない」「Tooltip の formatter で any 型を使っている」など
- **頼まれていない拡張（22%）**：「ダークモードやカラーバリエーションへの拡張性がない」「共通で使えるベースの型がない」など

- **守りの提案（8%）**：「データが空のときの表示がない」など。今の使い方では問題が起きないもの
- **不具合（5%）と頼まれた機能の欠け（5%）**：動かせば確かめられるもの

つまり、意見で差し戻す上司が直させたことの9割は、動かしても良し悪しを確かめられないことだった。5.1節で守りの書き方が増えたのは、この「守りの提案」や「拡張の提案」に応えた結果と考えられる。

なお、2回目と3回目の見直しでは、不具合の割合が34%と29%に上がった（61件と17件）。1回目に直させた結果として、不具合が入った可能性がある。実際、Bのある回（社員の表）では、1回目に「並べ替えの印が古く見える」「ダークモードの余白」「型のプロパティが少なすぎる」と差し戻し、4つのファイルがすべて書き直された。最後の版は、プレビューでエラーになって動かなかった。それでも2回目の Ren は95点で合格を出していた。動かさない上司には、自分の直させた結果が壊れたことが見えない。

7 確かめの差し戻し

確かめの差し戻しが実際に起きたのは、Cの48回中3回（bmi・landing・mortgage、合わせて4回の直し）だけだった。3回とも、最後は動いて、審査で合格した。Dでは一度も起きなかった。今のYUAのチームは、動かないアプリをほとんど作らない（Aで47/48が動いた）ため、確かめが鳴る場面が少ない。

鳴る場面が少ないので、確かめの差し戻しの効き目は、統計ではっきりとは言えない（合格+6.2ポイント、 $p=0.16$ ）。しかし、かかったものもほとんどなかった。AとCを比べると、料金は100題あたり\$1.65と\$1.70、時間の中央値はどちらも37秒で、動いた回は47/48から48/48になった。確かめの差し戻しは、**必要**なときだけ鳴り、**鳴ったときは直す**。意見の差し戻しが、毎回料金を3.6倍にして、何も直さないことがあったのと対照的である。

8 結果の確かさ

- **審査役の一致**：2人の審査役の Krippendorff の α は、総合点で 0.93、観点ごとに 0.76（簡潔さ）～0.93（見た目）だった。元の論文（0.08～0.30）よりずっと高い。ただし2人とも同じ Gemini の系統で、好み
が似ているだけかもしれない
- **審査役を1人ずつ外す**：AとBの総合点の差は、Gemini 3.8 Flash だけだと +0.00（ $p=0.98$ ）、Gemini 3.7 Flash だけだと +0.10（ $p=0.43$ ）で、どちらも結論は変わらない
- **合否の審査**：5つの観点の審査とは別の、合否の審査（3回の真ん中）でも、意見の効果は +0.04（0～2点、 $p=0.53$ ）、確かめは +0.04（ $p=0.44$ ）だった。ロジスティック回帰（お題ごとにまとめた標準誤差）でも、どの項も $p>0.3$
- **検出力**：48組の対応のある比べは、 $d_z \approx 0.41$ の差を80%の確率で見つけられる。元の論文の主な差（ $d=0.34\sim0.42$ ）と同じくらいの大きさであり、元の論文（43組）とほぼ同じ規模である
- **天井**：点は4.6～4.8と高く、上に余裕が少ない。元の論文（4.4～4.7）と同じ事情である

8.1 人の目による確かめ

元の論文が人による確かめをしたのなら、著者が、差し戻しのない形 A と、意見の差し戻しだけの形 B を、お題ごとに1組ずつ（24組）見比べた。組は、同じお題・同じ回で、両方ともアプリができた回から乱数で選んだ。2つのアプリは、本番と同じ仕組みで組み立てた、実際に操作できる画面として左右に並べ、どちらの形かは伏せ、左右も乱数で入れ替えた。両方をさわってから「頼んだ人にとってどちらが良いか」を5段階（-2～+2）で選び、理由をひとこと書いた。コードと審査役の点は見えていない。

人の判断	組の数
B（意見の差し戻しあり）が良い	11
同じくらい	5
A（差し戻しなし）が良い	8

表6 人の見比べ（24組、どちらの形かは伏せた）。お題ごとに1組。

平均は B の向きに +0.21（ブートストラップの95%区間 -0.21～+0.62、Wilcoxon の $p = 0.31$ 、符号検定 $p = 0.65$ ）で、偶然の範囲である。A がエラーで動かなかった1組を除くと、10組対8組（+0.13、 $p = 0.51$ ）になる。同じ24組で、AI の審査役の差も +0.05（ $p = 0.85$ ）だった。人の判断と審査役の点の差は、ほどほどにそろっていた（順位相関 $\rho = 0.48$ 、 $p = 0.017$ 。動かなかった1組を除くと $\rho = 0.40$ 、 $p = 0.057$ ）。ただし、1組ずつで向きが合ったのは24組中12組で、審査役の点は、まとめて見れば当てになるが、1つずつの判定には使えない。

書かれた理由からは、使う人が何に気づくかが見える。差のついた19組のうち12組は、片方にしかない機能（休憩タイマー・元に戻す・例文・地図・返済方式の比べ）が決め手で、その向きは両方にあった（B が7組、A が5組）。残りは、見た目や読みやすさが4組、不具合（エラー、クイズの解説に別の言語の文字が混じる、ドラッグで動かない）が3組だった。B を良くした機能が、直しで足されたものか、最初の版からあったものかは調べていない。

9 考察

9.1 元の論文と、何が同じで、何がちがうか

同じだったのは、仕組みである。意見で差し戻せる上司がいても、最初の版は変わらない。差が生まれるのは直しで、直すたびに中身が足され、確かめられない心配への備えが増えた。文章ではぼかしの言葉が、コードでは守りの書き方が増える。上司が直させたことの9割は、動かしても確かめられない意見だった。

ちがったのは、その結果である。元の論文では、直して報告書の点が下がった。本実験では、コードの点は下
がらず、簡潔さだけが下がる向きにあった（偶然の範囲）。考えられる理由は2つある。1つは、コードを作る
仕事では、足されたものの多く（スマホ対応・空のデータの表示・型の追加）が、害にならないことである。
報告書のぼかしは読む人の判断を鈍らせるが、コードの守りの書き方は、使う人には見えない。もう1つは、
作り手が直すたびに、アプリが動くという強い制約がかかっていることである。

9.2 「足を引っ張る」より「お金と時間を使う」

本実験で、意見で差し戻す権限が払わせたのは、出来ではなく、料金（3.6倍）・時間（+29秒）・コードの量
（+27%）だった。出来が変わらないのに料金が3.6倍になるなら、それは足を引っ張っているのと同じことで
ある。元の論文の推測（「上司は確かめられるときだけ役に立つ」）は、出来の面では確かめられなかったが、
かかったものと得たものの比で見ると、強く支持される。確かめの差し戻しは、ほとんど何も払わずに、動か
ない回を直した。意見の差し戻しは、多くを払って、確かめられる何かを直したとは言えない。

9.3 YUA への示唆

本番の YUA の Ren は甘く、ほとんど差し戻さない（前回の測定で96回中6回）。これは結果として正しい設計
だった。Ren を厳しくすると、料金が数倍になるのに、出来は上がらない。差し戻しは、動かして確かめられ
るエラーに任せるほうがよい。Ren の見直しを残すなら、差し戻しのきっかけを、意見ではなく確かめられる
問題（エラー・頼まれた機能の欠け）に限るのがよい。これは元の論文の著者らの提案（「差し戻しは、上司
の判断ではなく、確かめ役の旗で起こすべき」）とも一致する。

10 限界

- **上司の校正**：意見の差し戻しを実際に起こすため、1回目は必ず改善点を挙げさせた（3.3節）。これは本番
の Ren とも、元の論文の上司ともちがう。自然に差し戻す上司では、結果がちがう可能性がある
- **確かめが鳴った回が少ない**：確かめの差し戻しは48回中3回しか起きなかった。確かめの効き目と、2つの
交互作用は、ほとんど測れていない。もっと難しいお題か、もっと弱い作り手で確かめる必要がある
- **人による評価は小さい**：2人の審査役はどちらも Gemini の系統で、一致が高いのは、好みが同じだからか
もしれない。作るモデルも Gemini である。人の確かめ（8.1節）は A と B の24組だけで、見比べたのは1
人、しかもこのシステムを作った著者本人である。どちらの形かは伏せたが、人どうしの一致は測れてお
らず、C と D は人が比べていない
- **1つのチーム・1つのモデル**：作り手も上司も Gemini 3.5 Flash-Lite だけで、元の論文のような複数のモデル
の組み合わせではない
- **仕事の種類**：日本語の小さなアプリ24題だけである。大きなコードや、長い時間をかける仕事では、直し
の効き目がちがう可能性がある
- **守りの書き方の数え方**：決まった一覧で数えているので、必要な備えと、むだな備えを見分けていない。
元の論文のぼかしの言葉の数え方と同じ限界である

- 指摘の振り分けは AI：2つの振り分けはよく一致したが、どちらも Gemini で、人は確かめていない
- 差し戻しの回数：2回目の差し戻しは上司の判断で起きるので、回数と出来の関係は原因と結果ではない

11 まとめ

Agachan ほか [1] の「上司は、確かめられるときだけ役に立つ」という推測を、本番のコードを作る AI のチームで、意見の差し戻しと確かめの差し戻しを分けた2×2の実験で確かめた。192回の結果では、どちらの差し戻しも、出来を偶然の範囲を超えて変えなかった。元の論文とちがい、意見の差し戻しで出来は下がらなかった。

しかし、意見の差し戻しは、料金を3.6倍、時間を1.7倍、コードを1.27倍にした。直しのたびにコードは膨らみ、守りの書き方が増えた。上司の指摘の9割は、動かしても確かめられない意見だった。一方、動かして確かめて差し戻す仕組みは、必要な3回だけ鳴り、3回とも直し、ほとんど何も払わなかった。

確かめられない意見で直させる権限は、少なくともこの仕事では、出来を壊すのではなく、出来を変えずにお金と時間を使う。差し戻しのきっかけは、上司の意見ではなく、確かめられる問題に結びつけるのがよい。

参考文献

1. B. Agachan, M. van Duijn, A. Zohrehvand. “Loop-Back Authority in LLM Agent Teams: A Paired Experiment on Flat and Hierarchical Coordination.” arXiv:2609.14767v2, 2026.
2. 湯村 信司. 「AIのチームは、どこで失敗するか」 . AQUA Research Technical Report 005, 2026.
3. 湯村 信司. 「1人のAIで十分か」 . AQUA Research Technical Report 004, 2026.
4. 湯村 信司. 「AIが作ったアプリを、AIの審査で正しく測れるか」 . AQUA Research Technical Report 002, 2026.
5. K. Krippendorff. *Content Analysis: An Introduction to Its Methodology*. 4th ed., SAGE, 2018.

データ：お題・4つの形の結果（2つの実験の全384回）・作られたアプリと画面・返事の記録・審査の点・指摘の振り分け・人の見比べ・集計のプログラムを、GitHub で公開し、Zenodo (doi:10.5281/zenodo.23423127) に保管した。公開したデータで集計をやり直すと、本稿の数字がすべて同じになる。集計は `analyze-p6.py`、5つの観点の審査は `judge-rubric.cjs`、指摘の振り分けは `classify-critiques.cjs`。