

Testing a case preparation assistant

AI Evaluation and Governance Lab

I used Codex to build an AI case preparation app with Microsoft Foundry and synthetic healthcare administrative records. It tests whether a model can organize evidence for staff review. A holdout failure exposed a gap that the first test set missed and shaped the next product decision: expert review before a staff pilot.

Product scope and approach

The assistant summarizes a case, identifies missing documents and identity conflicts, and suggests a next step for a person to review. Foundry hosts GPT-5-mini. The app runs eight rule checks and puts prompt comparisons, source evidence, and human assessments in one workbench. A separate Cases area supports manual case tasks.

The Runs page offers explicit model-run controls, progress, cancellation, and cost estimates. Governance collects saved assessments. All six pages include an optional Walkthrough that explains useful sections. Short action labels and expandable details keep the working screens focused.

Observed evidence

Measure	Baseline	Grounded prompt
Development cases passing all eight checks	24 of 24	24 of 24
Holdout cases passing all eight checks	11 of 12	12 of 12
Median holdout request latency	18.3 seconds	18.8 seconds

72 outputs across 36 authored cases, one attempt per prompt and case. Pass counts measure the configured rules; they do not estimate accuracy. Latency includes adapter overhead.

What the failure changed

In holdout case 31, the baseline acknowledged another patient’s document but reported no identity mismatch. It recommended reviewing eligibility instead of resolving the conflict. The grounded prompt corrected both fields, but its summary grew from 49 to 173 words. Both passed exact-quote checks. This case showed why the product needs checks for required actions alongside a person’s judgment of claim support and readability.

Evaluator limits and the next decision

Foundry groundedness scored two selected case 18 summaries at 2 of 5 and 4 of 5, using the same deployment as the generator. Its explanations criticized omitted source content, including malicious instructions. That calls for calibrating the judge with domain experts; two selected outputs cannot establish a reliable prompt winner.

Next, I would have domain experts assess claim support, uncertainty, and usefulness, then compare correctness, correction effort, and time per task with the existing process. Human assessments and staff time savings remain unmeasured. All 73 generation calls and two judge calls used about \$0.33 in estimated model tokens, excluding taxes and other services.

Sources: [Recorded findings and reports](#) | [Microsoft Foundry groundedness guidance](#)