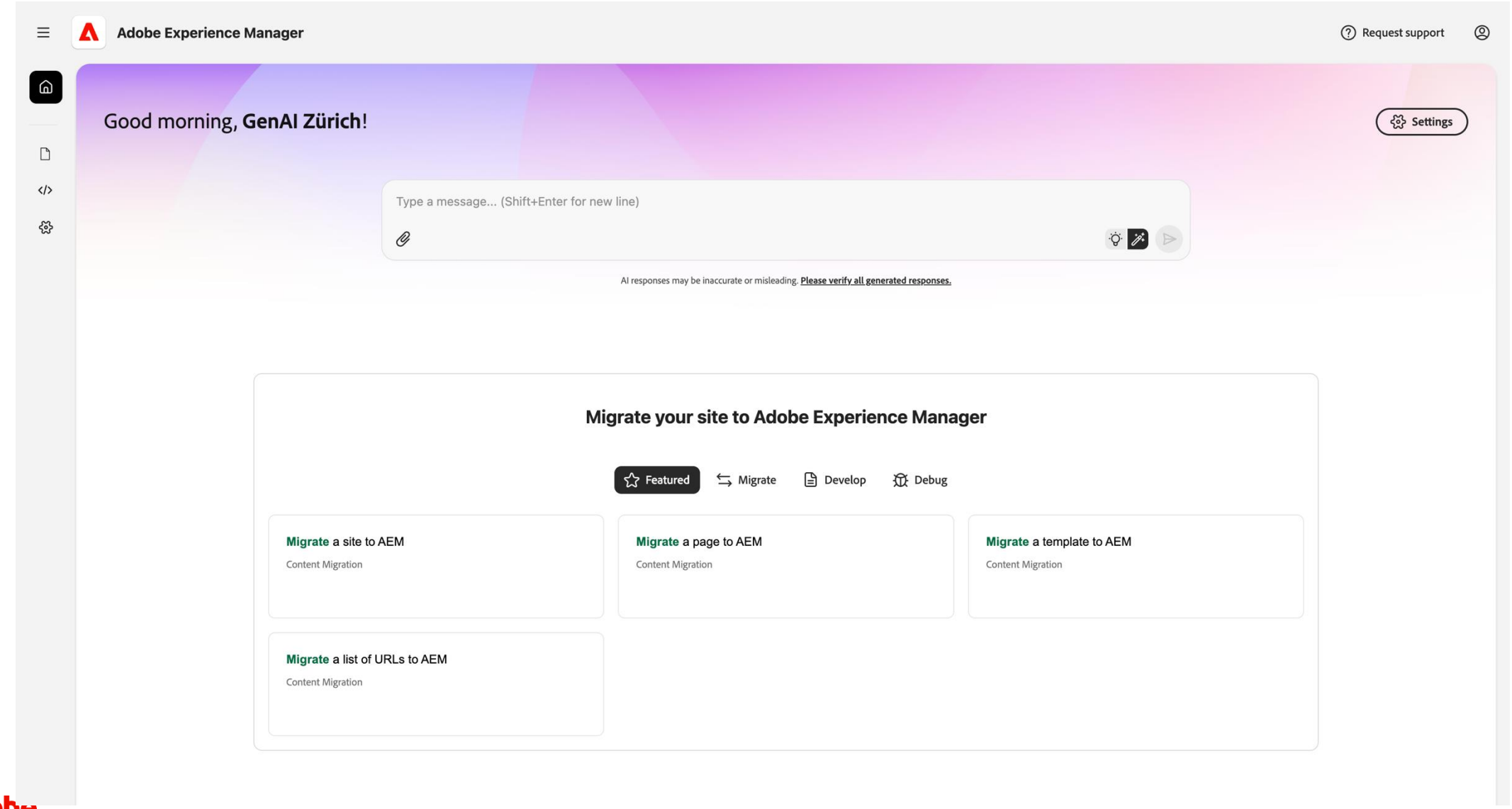
Testing LLM Outputs: Caging the Wind or Just Another Day in the Office?

Alex Carol | Software Development Engineer

AEM Growth & Adoption Team

Adobe

The subject under test



The subject under test

- **AEM Experience Modernization Agent**: AI-powered code migration for Adobe Edge Delivery Services
- Claude orchestrates 27 skills, subagents, and MCPs from a single prompt
- One prompt in, full website migration out
- Powerful... but how do you test something this complex?

The problem

- **The Butterfly Effect**: one change ripples everywhere
- 27 interconnected skills, each orchestrating others. Touch one, break many
- As the team grew, regressions started slipping through
- Manual testing couldn't keep up with the complexity

Adobe

We needed a testing strategy.

Our first attempt: code is cheap

- In the AI era, writing code is easy, so we just built a test framework
- Custom Node.js harness with Claude Agent SDK
- JSON test definitions, file assertions, LLM-as-judge scoring
- Over 500 lines of custom orchestration code
- Got it working in a few days

The struggle was real

- No caching, every rerun cost real money
- No visual UI to compare results across runs
- New team members couldn't figure out how it worked
- Tests passed locally, failed in CI, nobody knew why
- More time debugging the framework than writing actual tests

What actually worked

- **Promptfoo**: open-source LLM evaluation framework
- YAML configs replaced hundreds of lines of custom code
- Built-in caching, visual comparison UI, CI/CD integration
- We contributed Claude Agent SDK plugin support upstream
- Enabled safe model upgrades: swap Claude Sonnet for Opus, run the suite, compare scores

87 commits: what we learned

- One PR, 87 commits to migrate from custom to Promptfoo
- The architecture emerged through experimentation:
  - **Assertions**
  - **Evaluators**
  - **Hooks**
  - **YAML configs**
- **LLM outputs are non-deterministic: test for quality, not exact matches**

A real test case

```
- description: Migrate WKND homepage to EDS
  vars:
    testPrompt: >
      Migrate wknd-trendsetters.site to EDS
  assert:
    - type: javascript
      value: file://assertions/file-exists.js
      config:
        filePath: content/index.md
    - type: javascript
      value: file://evaluators/content-similarity
      threshold: 0.7
      config:
        original_url: https://wknd-trendsetters.site
        migrated_path: content/index.md
```

Claude evaluating Claude

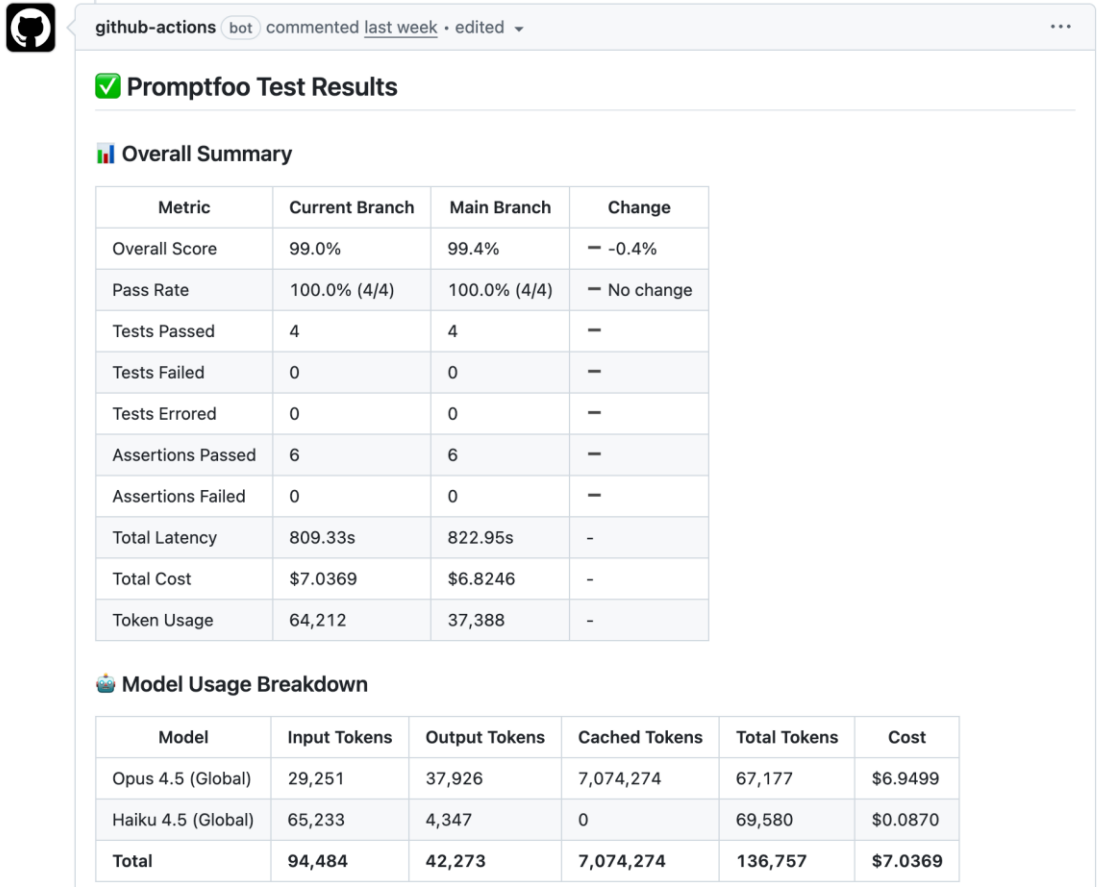
```
evaluators/content-similarity/
├── EVALUATOR.md
└── index.js

EVALUATOR.md tells Claude:
  1. Fetch the original website
  2. Read the migrated markdown
  3. Score the quality

  1.0  = Perfect
  0.7+ = Acceptable
  <0.7 = Needs work
```

Adobe

GitHub PR reports



github-actions bot commented last week • edited

✅ Promptfoo Test Results

📊 Overall Summary

Metric	Current Branch	Main Branch	Change
Overall Score	99.0%	99.4%	➖ -0.4%
Pass Rate	100.0% (4/4)	100.0% (4/4)	➖ No change
Tests Passed	4	4	➖
Tests Failed	0	0	➖
Tests Errored	0	0	➖
Assertions Passed	6	6	➖
Assertions Failed	0	0	➖
Total Latency	809.33s	822.95s	-
Total Cost	$7.0369	$6.8246	-
Token Usage	64,212	37,388	-

🤖 Model Usage Breakdown

Model	Input Tokens	Output Tokens	Cached Tokens	Total Tokens	Cost
Opus 4.5 (Global)	29,251	37,926	7,074,274	67,177	$6.9499
Haiku 4.5 (Global)	65,233	4,347	0	69,580	$0.0870
Total	**94,484**	**42,273**	**7,074,274**	**136,757**	**$7.0369**

📝 Individual Test Results

Test	Overall Score	Status	content-similarity
Migrate WKND Trendsetters homepage to EDS	0.96	✅ Pass	0.92 ➖ (-0.03)
[META] Evaluator: content-similarity	1.00	✅ Pass	1.00 ➖
[META] Assertion: file-exists	1.00	✅ Pass	-
[META] Plugins: excat and edge-delivery-services enabled	1.00	✅ Pass	-

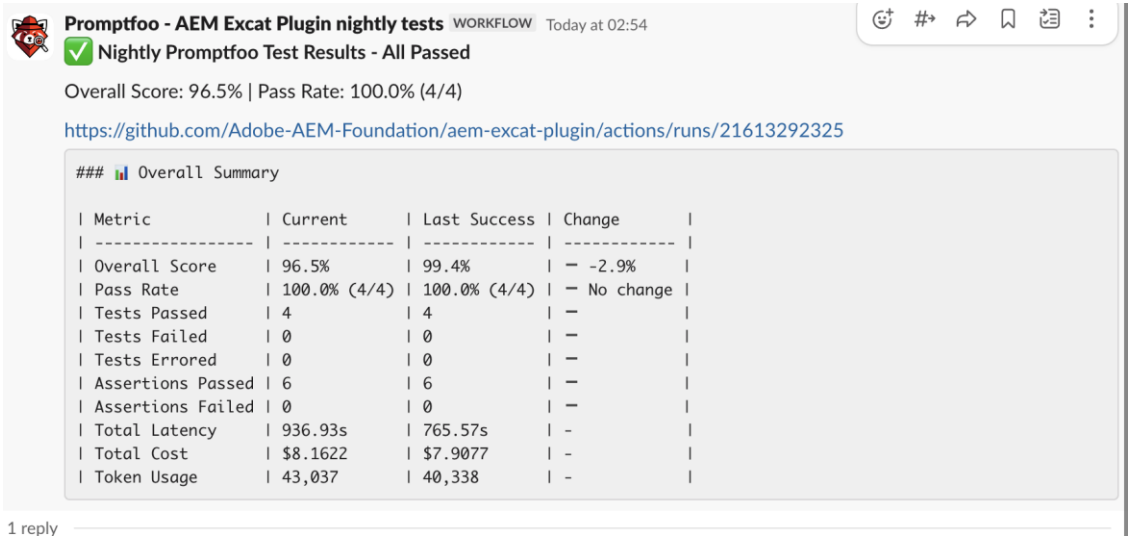
📈 Score Changes vs Main Branch

Test	Current Branch	Main Branch	Change
Migrate WKND Trendsetters homepage to EDS	0.96	0.97	➖ -0.02
[META] Evaluator: content-similarity	1.00	1.00	➖ No change
[META] Assertion: file-exists	1.00	1.00	➖ No change
[META] Plugins: excat and edge-delivery-services enabled	1.00	1.00	➖ No change

Legend

- ⬆️ Improvement vs baseline
- ⬇️ Regression vs baseline
- ➖ No change or low variation
- 🆕 New test/feature
- ❌ Test failed
- ✅ Test passed

Slack notifications



Promptfoo - AEM Excat Plugin nightly tests WORKFLOW Today at 02:54

✅ Nightly Promptfoo Test Results - All Passed

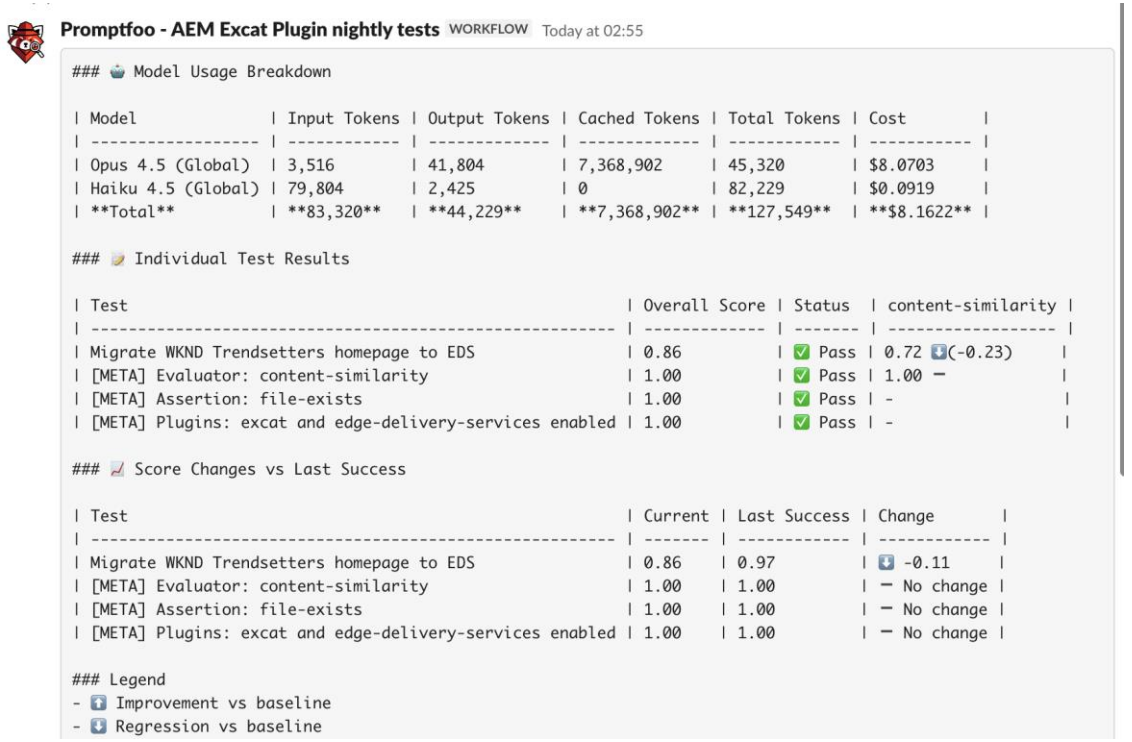
Overall Score: 96.5% | Pass Rate: 100.0% (4/4)

https://github.com/Adobe-AEM-Foundation/aem-excat-plugin/actions/runs/21613292325

📊 Overall Summary

Metric	Current	Last Success	Change
Overall Score	96.5%	99.4%	➖ -2.9%
Pass Rate	100.0% (4/4)	100.0% (4/4)	➖ No change
Tests Passed	4	4	➖
Tests Failed	0	0	➖
Tests Errored	0	0	➖
Assertions Passed	6	6	➖
Assertions Failed	0	0	➖
Total Latency	936.93s	765.57s	-
Total Cost	$8.1622	$7.9077	-
Token Usage	43,037	40,338	-

1 reply



Promptfoo - AEM Excat Plugin nightly tests WORKFLOW Today at 02:55

🤖 Model Usage Breakdown

Model	Input Tokens	Output Tokens	Cached Tokens	Total Tokens	Cost
Opus 4.5 (Global)	3,516	41,804	7,368,902	45,320	$8.0703
Haiku 4.5 (Global)	79,804	2,425	0	82,229	$0.0919
Total	**83,320**	**44,229**	**7,368,902**	**127,549**	**$8.1622**

📝 Individual Test Results

Test	Overall Score	Status	content-similarity
Migrate WKND Trendsetters homepage to EDS	0.86	✅ Pass	0.72 ⬇️(-0.23)
[META] Evaluator: content-similarity	1.00	✅ Pass	1.00 ➖
[META] Assertion: file-exists	1.00	✅ Pass	-
[META] Plugins: excat and edge-delivery-services enabled	1.00	✅ Pass	-

📈 Score Changes vs Last Success

Test	Current	Last Success	Change
Migrate WKND Trendsetters homepage to EDS	0.86	0.97	⬇️ -0.11
[META] Evaluator: content-similarity	1.00	1.00	➖ No change
[META] Assertion: file-exists	1.00	1.00	➖ No change
[META] Plugins: excat and edge-delivery-services enabled	1.00	1.00	➖ No change

Legend
- ⬆️ Improvement vs baseline
- ⬇️ Regression vs baseline

Adobe

The cost problem

- LLM tests consume **tokens**, and they add up fast
- Full test suite = multiple Claude conversations per test case
- Running on every PR? Unsustainable

We needed a smarter approach.

How we made it sustainable

Nightly runs

Full suite runs overnight
Fresh baseline every morning

On-demand testing

PR label triggers tests
Test only when changes matter

Caching

Skip unchanged tests
Avoid redundant token spend

Token tracking

Per-test cost visibility
Find and optimize expensive tests

Adobe

Key takeaways

🎲 **Embrace non-determinism**: test for quality, not exact outputs

🔍 **LLM-as-judge works**: structured criteria produce reliable quality scores

🛠️ **Know when to build vs adopt**: Promptfoo saved us months of framework work

💰 **Track your token spend**: visibility into costs lets you optimize what matters

Questions?

alexcarol.com/talk/testing-llm-outputs/