ILLUSTRATIVE EXAMPLE — NOT A REAL CUSTOMER ENGAGEMENT

This is a mocked Week 4 readout from a hypothetical engagement. All numbers, quotes, and team names are synthetic. Provided so prospects can see the artifact shape before committing to a sprint.

SAMPLE WEEK 4 READOUT

AI Context Sprint Lite — Platform Reviews Team

Fictional engagement at SampleCo · workflow: code_review

Parameter	Value
Customer (fictional)	SampleCo — Platform Reviews Team (8 engineers)
Workflow under measurement	code_review
Sprint window	30 days, four working weeks
Baseline window	30 days prior to sprint start
Deployment mode	Vetratek-managed Fly.io workspace; MemFlair-managed Anthropic Claude
Sources in scope	acme-platform-api repo (PRs, reviews, conventions); API Review Conventions doc; postmortem index
Operator	Vetratek delivery lead (fictional)

Executive summary

The Platform Reviews Team ran the code_review workflow on MemFlair-prepared bundles for four weeks. Cycle time, time to first reviewer response, and revision rounds all moved in the expected direction. Adoption was broad — 7 of 8 reviewers used bundles, with the 8th joining the team mid-sprint. Bundle relevance ratings averaged 4.1 of 5 across 71 ratings. One material failure mode appeared in week 3 (stale-suppression false positives during an architecture document update) and was resolved within the sprint.

> *Recommended next step: extend MemFlair to the on-call rotation's incident response workflow. Retain code_review at current configuration. Re-evaluate freshness window settings in 60 days.*

Outcome metrics — did the workflow improve?

M1 **PR cycle time (open → merged)** · *Outcome*

Baseline: median 3.2 days **Pilot:** median 2.4 days **Direction:** **improved (~25%)**

Sample size: baseline n = 118 PRs · pilot n = 124 PRs

Bundles arriving with conventions and prior decisions at the top reduced reviewer onboarding time per PR, which compressed the wait phase before merge.

M2 **Time to first reviewer response** · *Outcome*

Baseline: median 8.5 hours **Pilot:** median 4.2 hours **Direction:** **improved substantially**

Sample size: baseline n = 118 PRs · pilot n = 124 PRs

The cleanest signal in the readout. Prepared context lowered the activation cost of starting a review; reviewers engaged faster because they no longer needed to find their footing on each PR.

M3 Revision rounds per PR · *Outcome*

Baseline: mean 2.3 rounds **Pilot:** mean 1.7 rounds **Direction:** **improved**

Sample size: baseline n = 118 PRs · pilot n = 124 PRs

Risk-flag context appearing in the bundle let authors anticipate review concerns; some second-round corrections moved into pre-PR self-review.

Usage metrics — did the team adopt the system?

M4 Bundle requests per PR · *Usage*

Baseline: n/a (system not deployed) **Pilot:** 0.82 requests/PR (102 bundles across 124 PRs)
Direction: **n/a — adoption floor met**

Sample size: pilot n = 124 PRs

Floor target was "≥ 60% of pilot-scope PRs had a bundle requested." Met on day 9 and held for the remaining 21 days.

M5 Unique reviewers using bundles · *Usage*

Baseline: n/a **Pilot:** 7 of 8 team members **Direction:** **n/a — broad adoption**

Sample size: team size 8

The 8th reviewer joined the team in week 3 and was onboarded into the workflow during their first review cycle.

M6 Prompt-kit usage rate · *Usage*

Baseline: n/a **Pilot:** 64% of bundles paired with a prompt-kit invocation **Direction:** **n/a**

Sample size: pilot bundles n = 102

Risk-flag analysis was the most-used prompt-kit slot. Prior-example surfacing was used less than expected; we will revisit the prompt for that slot in the next iteration.

Quality metrics — was the context useful and trustworthy?

M7 Bundle relevance rating · *Quality*

Baseline: n/a **Pilot:** average 4.1 across 71 ratings **Direction:** **n/a — above 3.5 floor**

Sample size: n = 71 ratings

Relevance rose from a week-1 average of 3.7 to a week-4 average of 4.3. The improvement tracks the context catalog growing through approved write-backs.

M8 Write-back candidate approval rate · *Quality*

Baseline: n/a **Pilot:** 47% approved or edited-and-approved (32 of 68 candidates) **Direction:** **n/a — within healthy band**

Sample size: n = 68 candidates

Healthy band is 30–70%. A very high rate would suggest a too-conservative candidate generator; a very low rate would suggest noisy candidates. We landed mid-band, which is what we want.

M9 Stale-suppression effectiveness · *Quality*

Baseline: n/a **Pilot:** 412 candidates suppressed as stale · 4.1% reviewer-confirmed false-positive rate **Direction:** **n/a — healthy**

Sample size: n = 412 suppressions · 17 false positives flagged by reviewers

The freshness model is active and accurate after tuning (see Failure mode 1 below). The 4.1% false-positive rate is within the band we consider acceptable for v1.

Qualitative reviewer feedback

Three reviewer quotes captured during the week 4 retrospective. Attributions are role-only; identities anonymized to the team. (All illustrative; no real customer.)

> *"I used to spend the first 20 minutes of every review reconstructing what we decided about this subsystem six months ago. The bundle has those decisions right at the top with the original PR linked."*
>
> *— Senior Reviewer, Platform Reviews Team*

> *"First sprint I've worked on where the readout matches what I actually saw in my day-to-day. The cycle-time number is real for our team, not a marketing average."*
>
> *— Platform Lead*

> *"We approved more review conventions in the last two weeks than the previous quarter. The write-back queue made it routine instead of a side project."*
>
> *— Engineering Manager*

Failure modes observed and resolved

1. Stale-suppression false-positive spike during architecture document update (week 3)

When a major architecture document was updated mid-sprint, the freshness model flagged the prior version stale faster than the new version was ingested and chunked. False-positive rate on suppressed candidates spiked from a steady 3% to 11% for ~36 hours. Three reviewers flagged missing context in that window.

Mitigation:

Freshness window for items typed architecture_note was extended from 14 to 30 days, allowing the new ingestion to land before the prior version was excluded. False-positive rate returned to the steady-state range within 48 hours of the change.

2. Bundle adoption dip in week 2 due to output formatting

Bundle request rate dropped roughly 40% in days 8–10 after one reviewer reported confusing section headers in returned bundles. Word spread within the team and several reviewers temporarily reverted to manual review.

Mitigation:

Section headers in the code_review workflow profile were tightened (clearer labels for "approved_conventions" vs. "prior_decisions"). Adoption recovered within 4 days and exceeded pre-dip levels in week 3.

3. Write-back candidates generated from an archived source repo

Two write-back candidates were generated from an archived sibling repo whose context the team no longer considered authoritative. Both candidates were rejected on review.

Mitigation:

Added the archived repo to the ingestion exclusion list. No further candidates from that source after the change.

Expansion recommendation

Three options were considered for the next expansion step. The recommendation is option B.

Option A: Add a second engineering team to code_review.

Lower risk, similar deployment. Trade-off: this is breadth, not depth, and doesn't validate MemFlair against a second workflow type. Reasonable but not the highest-value next move.

Option B (recommended): Extend MemFlair to incident response handoff workflow on the same team.

The Platform Reviews Team is also the on-call rotation for the same services. Incident response context (recent decisions, recent incidents, owner mapping, runbooks) overlaps materially with code_review context, so most of the catalog can be reused. A second workflow on the same team also tests whether MemFlair's review queue scales across workflow types without operator overhead.

Option C: Defer expansion; deepen code_review with custom retrieval profiles.

Conservative. Recommended only if budget or team capacity is constrained. The current code_review configuration is performing well enough that incremental tuning yields smaller returns than a second-workflow expansion.

> *Recommendation: pursue Option B. Scope a 4-week incident response extension at a reduced price (~60% of standard Sprint Lite) reflecting shared infrastructure and existing catalog. Re-evaluate freshness window settings in 60 days based on additional data.*

ILLUSTRATIVE EXAMPLE — NOT A REAL CUSTOMER ENGAGEMENT

This is a mocked Week 4 readout from a hypothetical engagement. All numbers, quotes, and team names are synthetic. Provided so prospects can see the artifact shape before committing to a sprint.

VETRATEK LLC · SBA-Certified SDVOSB · SAM Registered
matt@vetratek.ai · vetratek.ai