Asian Actuarial Conference 2026

Can an AI Agent Conduct an Experience Study?

Redefining the Actuary's Role Through Harness Engineering

19 August 2026

HyunSu Kim, Munich Re

Munich RE

HyunSu Kim, ASA (just call me HS)

Senior Actuarial Consultant, Regional Life & Health Pricing, Munich Re Singapore

Munich RE

- 15 years in life & health insurance and reinsurance in Korea
- Led experience studies and assumption setting at Munich Re Korea
- Lecturer on predictive analytics and AI, Institute of Actuaries of Korea
- Driving the integration of AI and advanced analytics into actuarial methodologies across APAC & MEA

Slides & more

Why I ran this experiment

The bottlenecks

Munich RE

New shape every time

Every data submission arrives with different structures, definitions, and granularity. The first week of every study goes to rediscovery, not analysis.

Checklists that only grow

Data issues rarely repeat in the same form. Checks keep being added, but rarely removed.

One-off fixes, no reuse

Each issue is resolved and documented differently for each cedant. The records pile up, but most stay case-specific.

Why I ran this experiment

The bottlenecks

Munich RE

New shape every time

Checklists that only grow

One-off fixes, no reuse

Case-by-case handling keeps automation out of reach

Every cedant needs its own treatment, so nothing becomes standard.
Time goes to manual checks, not analysis.

Why I ran this experiment

The bottlenecks

New shape every time

Checklists that only grow

One-off fixes, no reuse

Case-by-case handling keeps automation out of reach

Every cedant needs its own treatment, so nothing becomes standard.
Time goes to manual checks, not analysis.

What if an AI agent handled the case-by-case work?

Harness Engineering

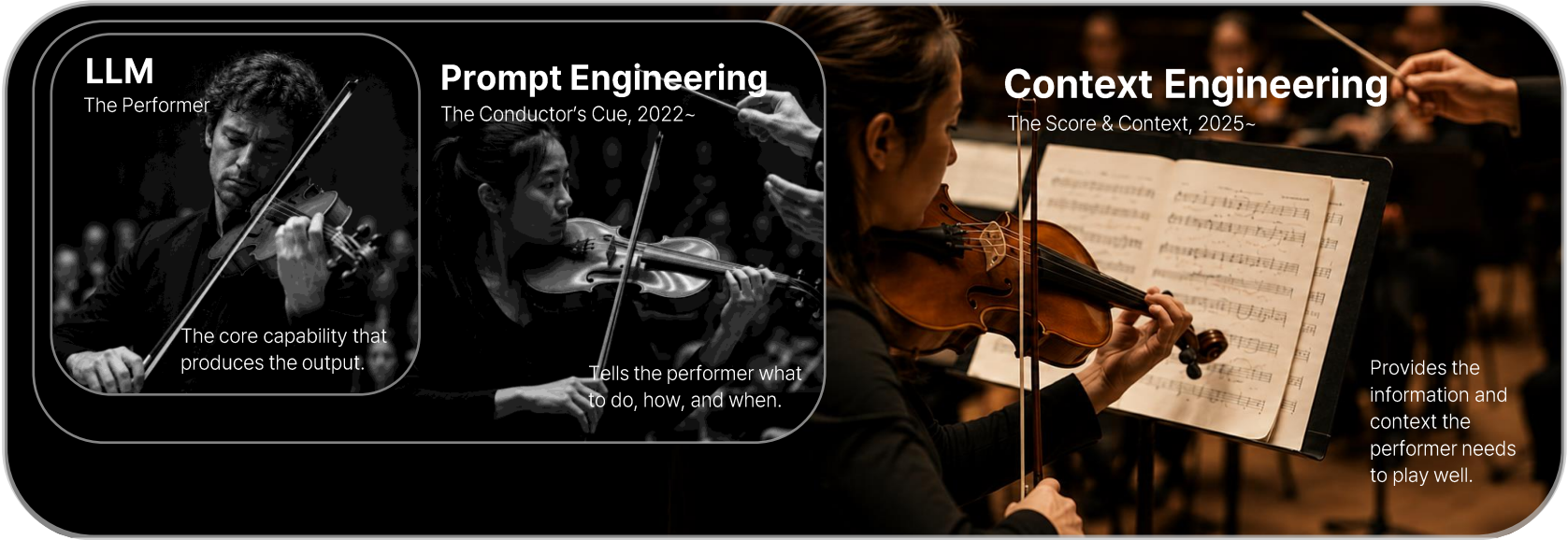
A term from the AI industry, early 2026

Harness Engineering

A term from the AI industry, early 2026

Harness Engineering

A term from the AI industry, early 2026

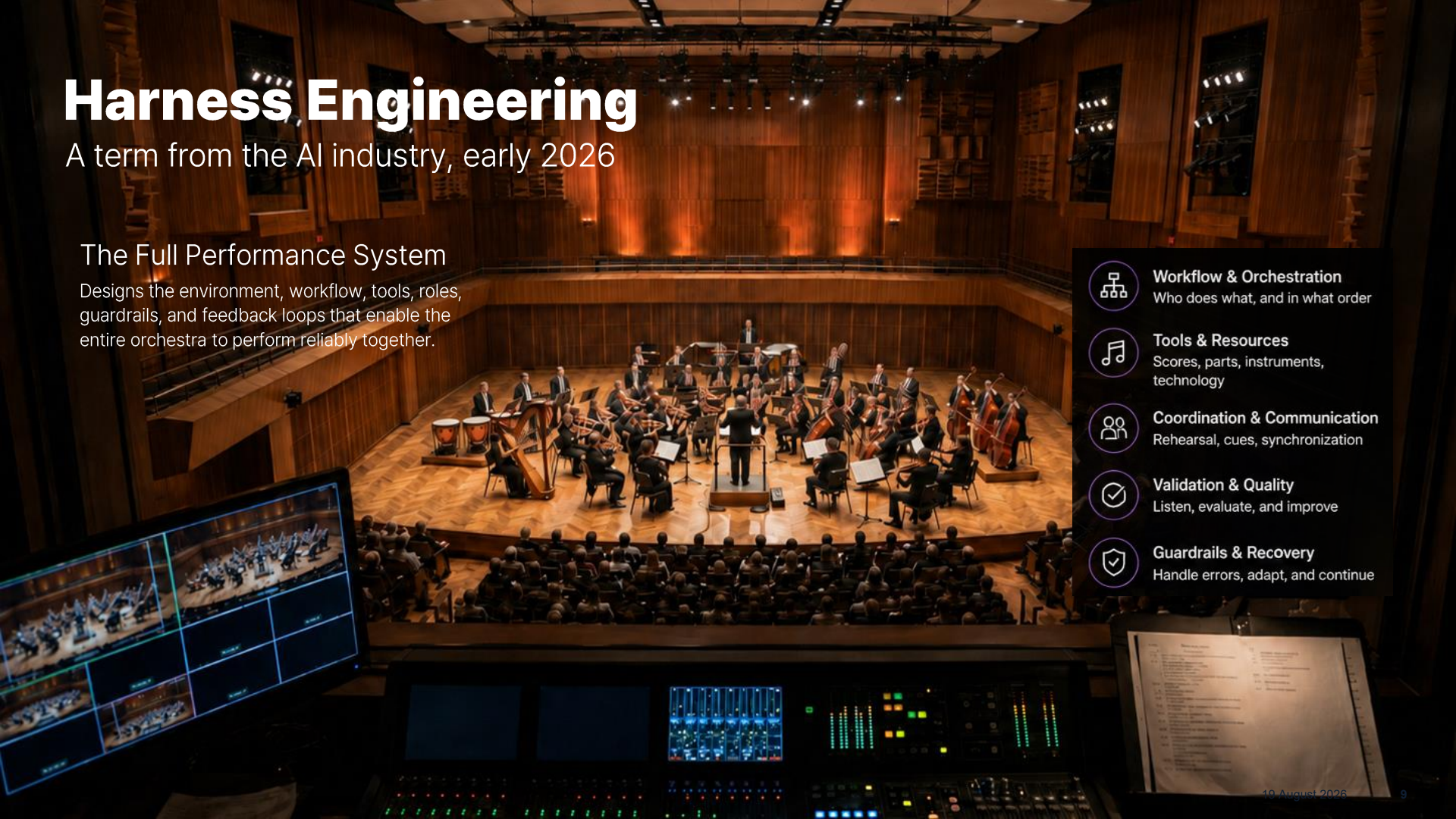


Harness Engineering

A term from the AI industry, early 2026

The Full Performance System

Designs the environment, workflow, tools, roles, guardrails, and feedback loops that enable the entire orchestra to perform reliably together.



Workflow & Orchestration
Who does what, and in what order

Tools & Resources
Scores, parts, instruments, technology

Coordination & Communication
Rehearsal, cues, synchronization

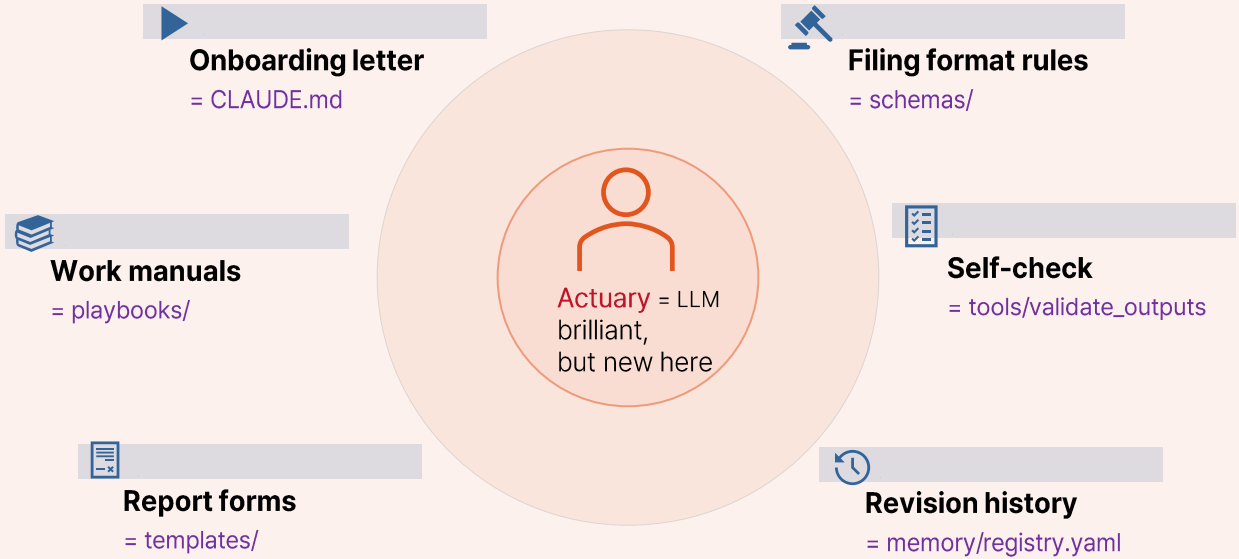
Validation & Quality
Listen, evaluate, and improve

Guardrails & Recovery
Handle errors, adapt, and continue

Harness Engineering

A term from the AI industry, early 2026

If the model is a talented new analyst, the Harness is everything the firm builds around them.

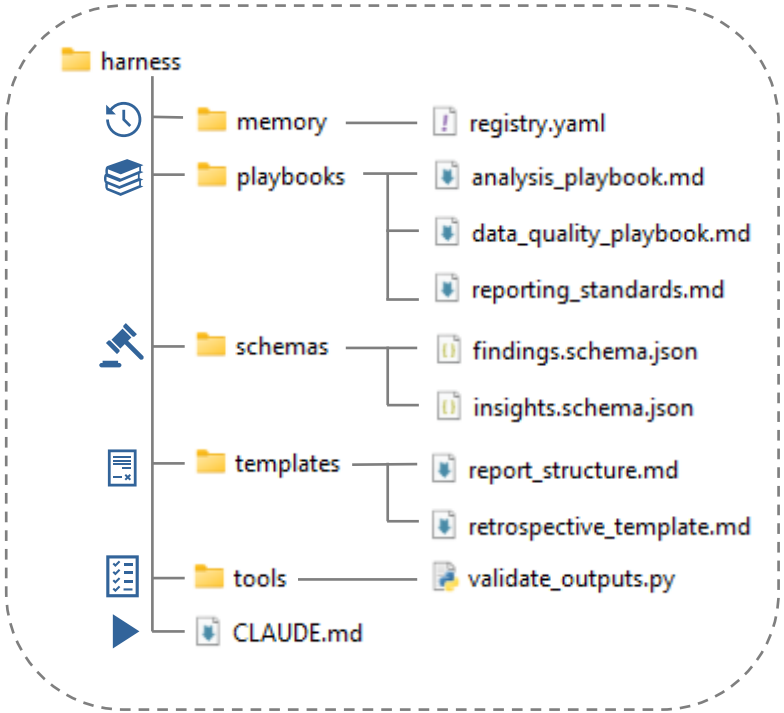


... plus an access badge that only opens certain doors (guardrails and isolation)

Inside the Harness

The harness is just text files

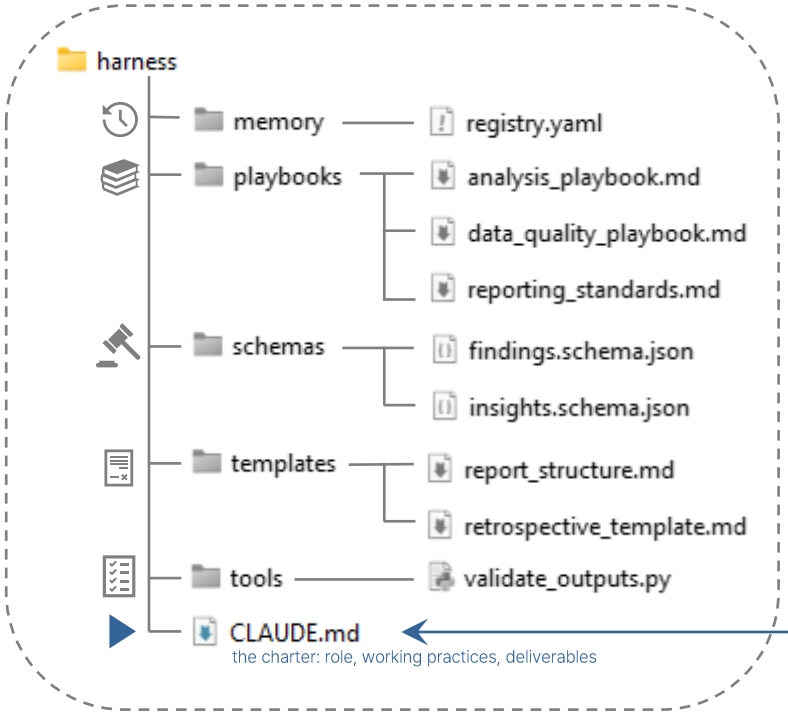
Munich RE



Inside the Harness

The harness is just text files

Munich RE



The charter: role, working practices, deliverables

CLAUDE.md

File Edit View

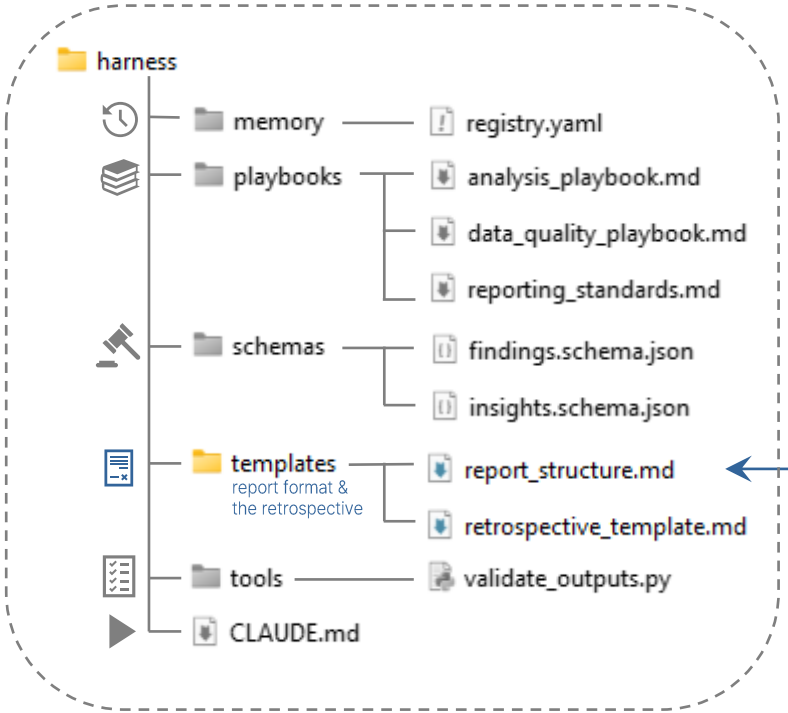
```
# Experience Study Workspace

## Who you are and what this is
You are the experience-study analyst at a life reinsurer. This workspace
contains a cedant's full-period data submission for a 100% quota-share YRT
treaty covering the cedant's Korean whole life and 5-year renewable term
portfolio, January 2019 – December 2025. The reinsurance premium in the
bordereaux is a monthly natural premium on the industry reference basis
(10th Korean experience life table reference net rates by attained age and
sex, applied to the ceded net amount at risk). Amounts are KRW thousands.

## Working practices
- Follow the playbooks in ./playbooks/ (data quality, analysis,
  reporting) during the corresponding phase of the work.
- Verify your own work. Before finalizing, reconcile: cleaned-data row
  counts against your findings; every number in the report against
  insights.json; exposure and claim totals before vs after cleaning, with
  the movement explained by your findings.
- Every material judgment call gets a written rationale — in findings.json
  for data decisions, in the report for analytical ones. "Material" means
  it could change a reported result.
- State the credibility of what you report. Where claim counts are thin,
```

Ln 17, Col 16 | 1,779 characters | Markdown s... | 100% | Unix (LF) | UTF-8

Inside the Harness

The harness is just text files

Munich RE



Experience Study Report Structure

report_structure.md

File Edit View

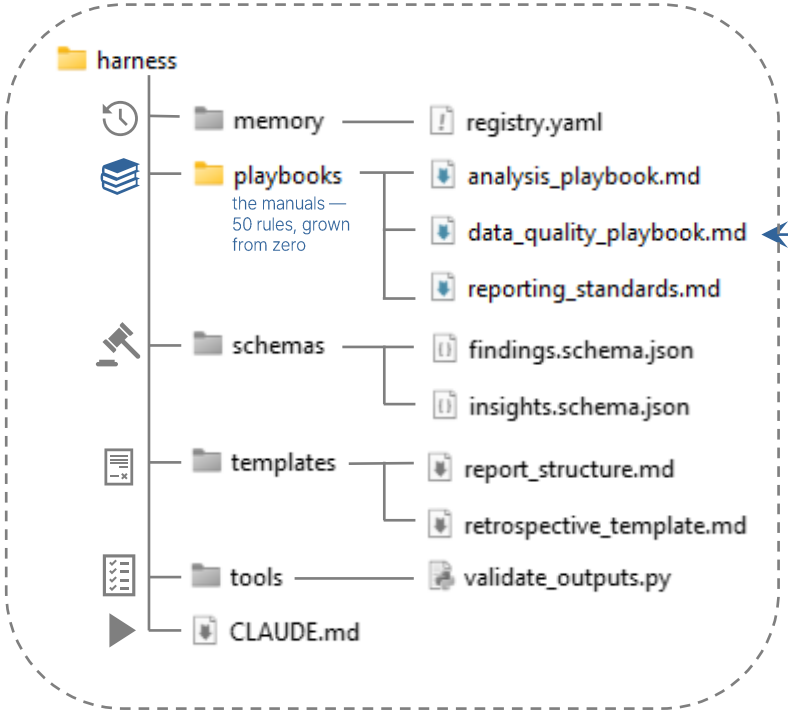
```
# Report Structure — Experience Study

Format contract for `./outputs/report.docx`. It fixes the section order and
presentation standards only; the methodology is yours to determine, apply,
and describe.

## Required sections, in this order

1. **Executive summary**
2. **Data received & quality assessment** — the issues you found, the
   treatment you applied to each, and the impact on exposure and claim
   totals (before vs after your cleaning).
3. **Methodology (as applied)** — describe what you actually did: exposure
   basis, expected basis, segmentation, credibility handling.
4. **Mortality experience results**
5. **Lapse experience results**
6. **Insights & recommendations**
7. **Cedant query list (appendix)** — all items requiring cedant
   confirmation, consolidated in one place.
8. **Limitations**

## Presentation standards
```

Ln 1, Col 1 | 1,265 characters | Markdown s... | 100% | Unix (LF) | UTF-8

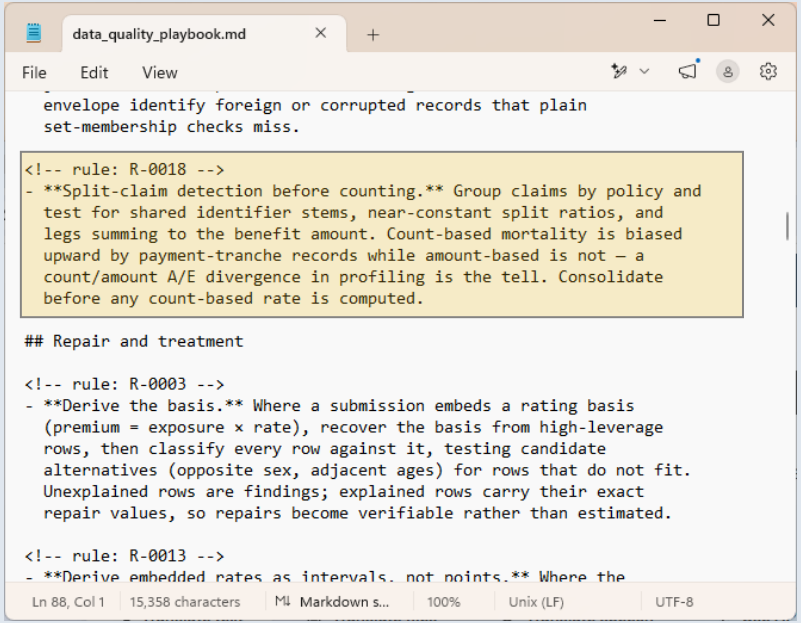
Inside the Harness

The harness is just text files

Munich RE



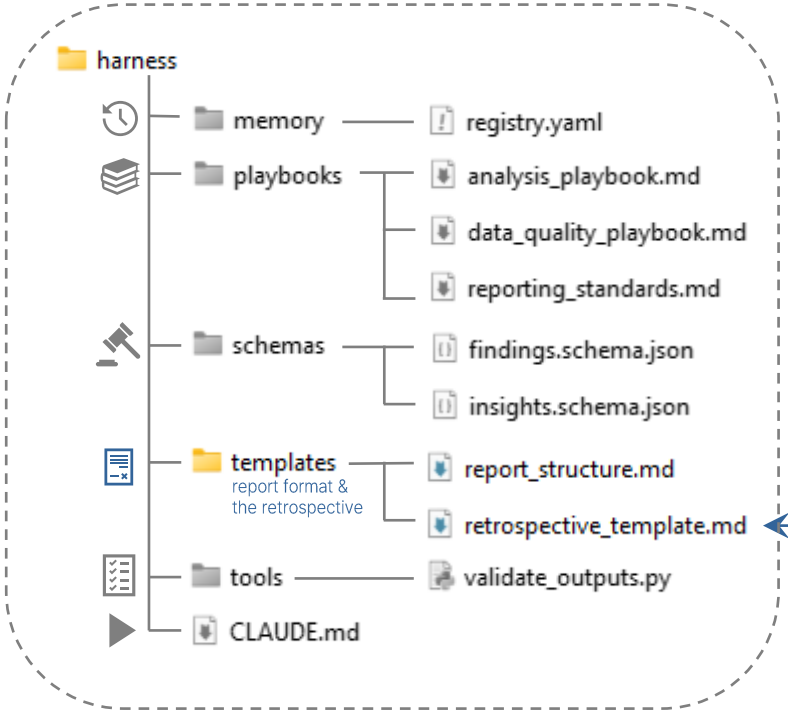
The work manuals: 50 rules, grown from zero



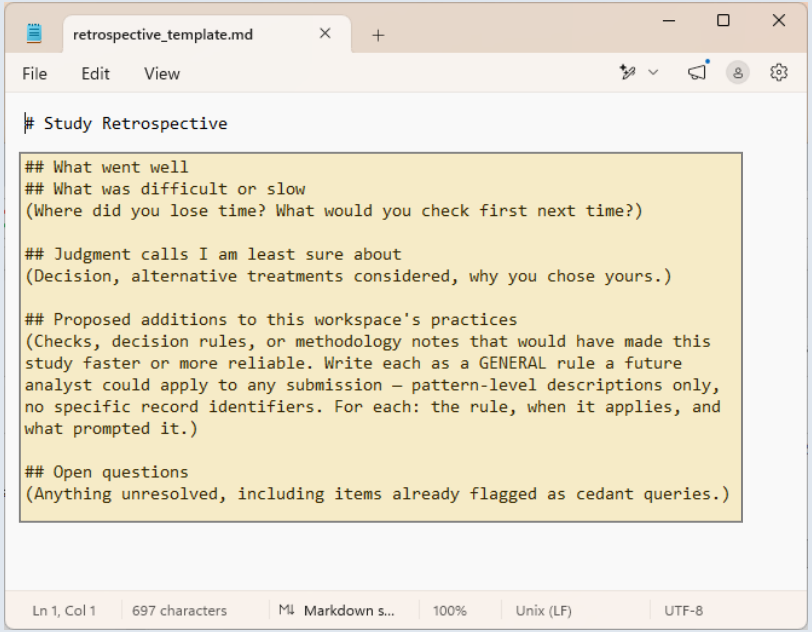
Inside the Harness

The harness is just text files

Munich RE



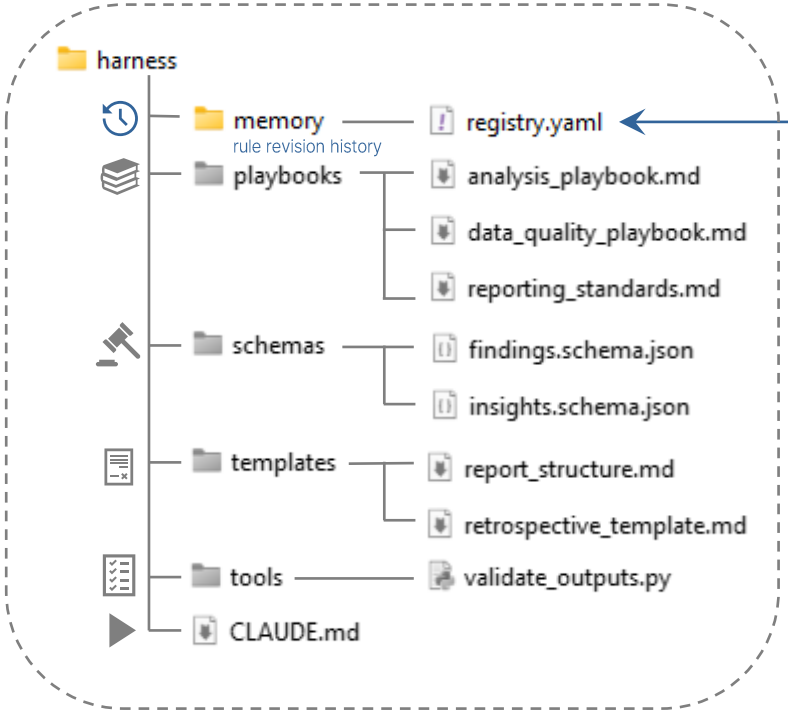
Retrospective template: where the loop feeds itself



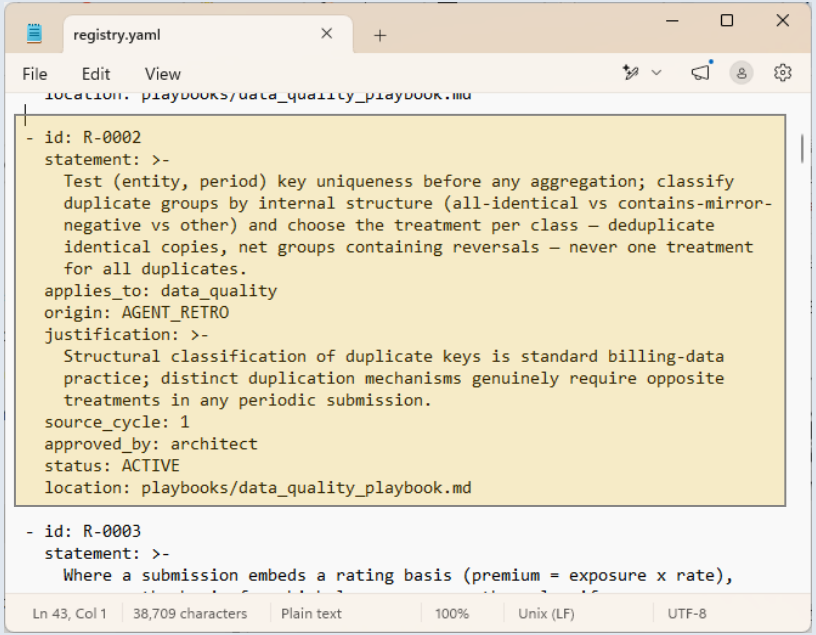
Inside the Harness

The harness is just text files

Munich RE

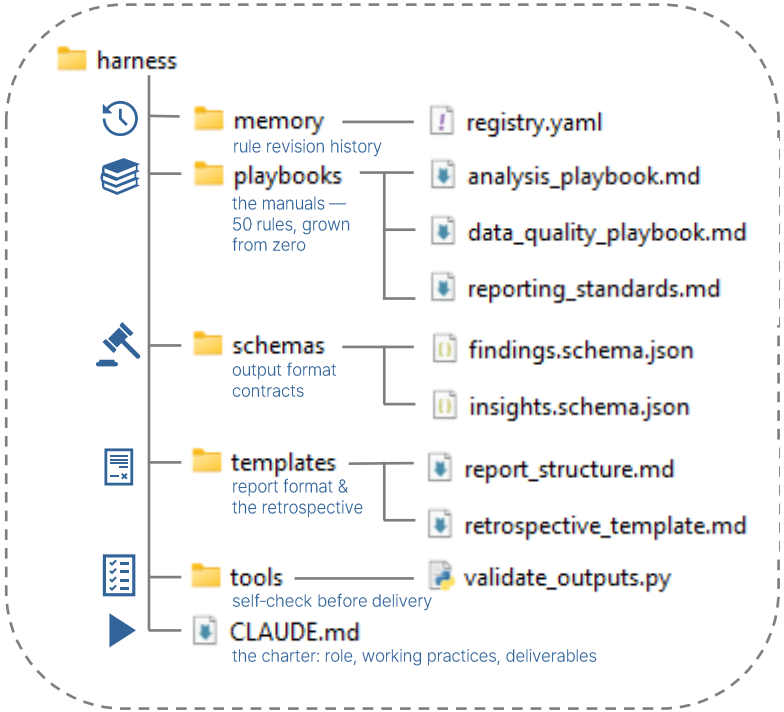


The memory: every rule's origin on record



Inside the Harness

The harness is just text files

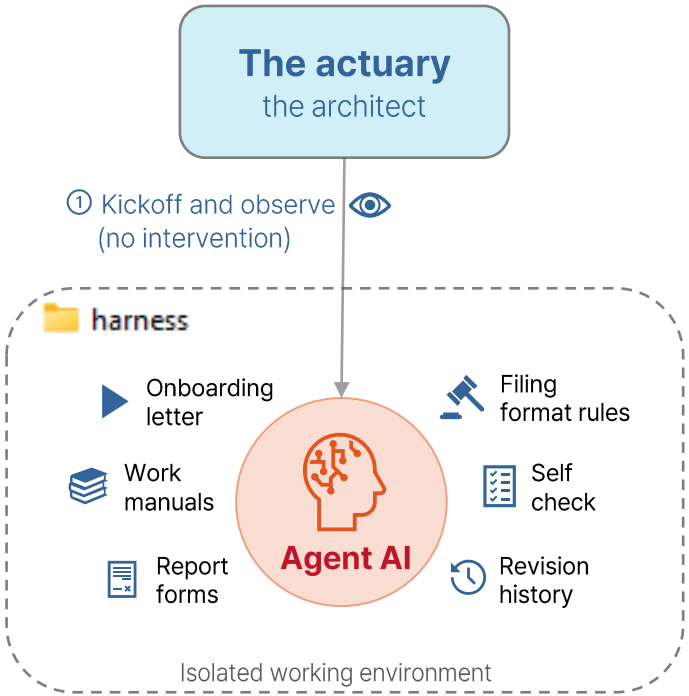


No code. No prompt tricks.

The same documentation we'd give a new actuary.

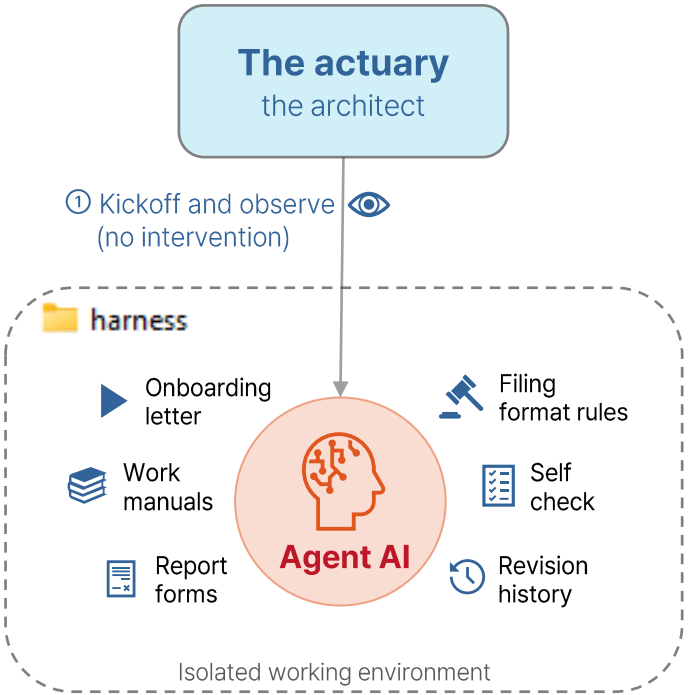
The study cycle

Human ON the loop — not in it
(for this experiment)



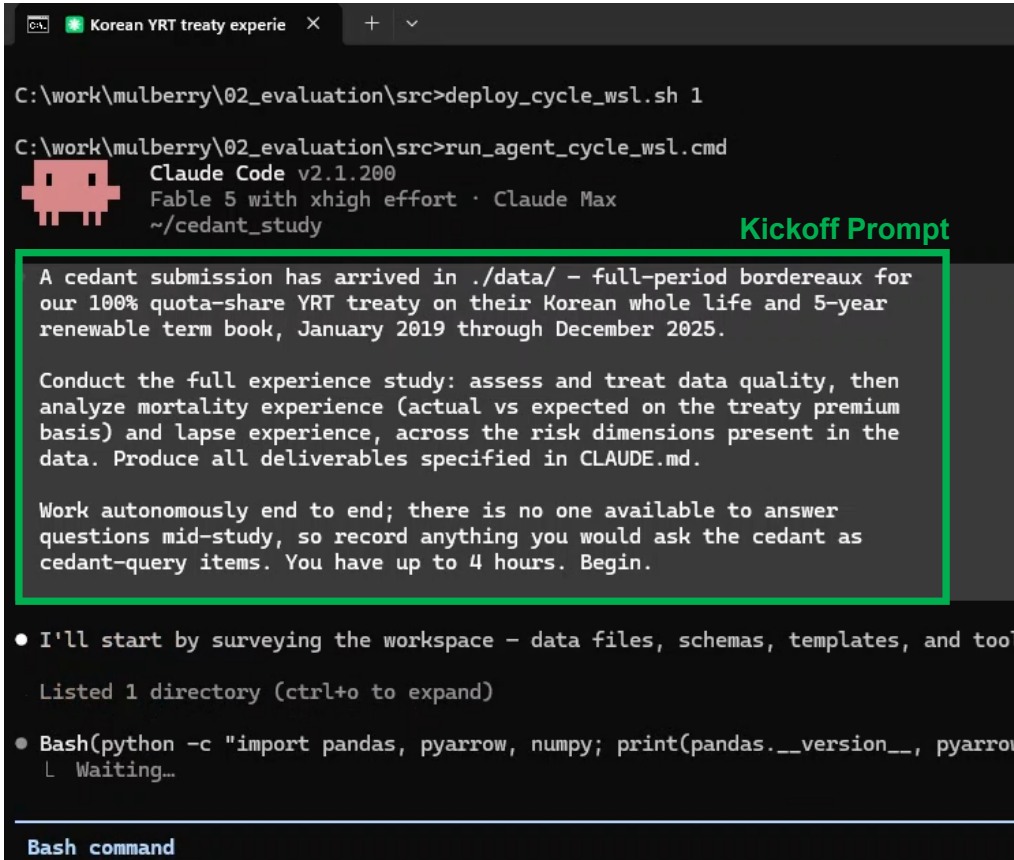
The study cycle

Human ON the loop — not in it
(for this experiment)



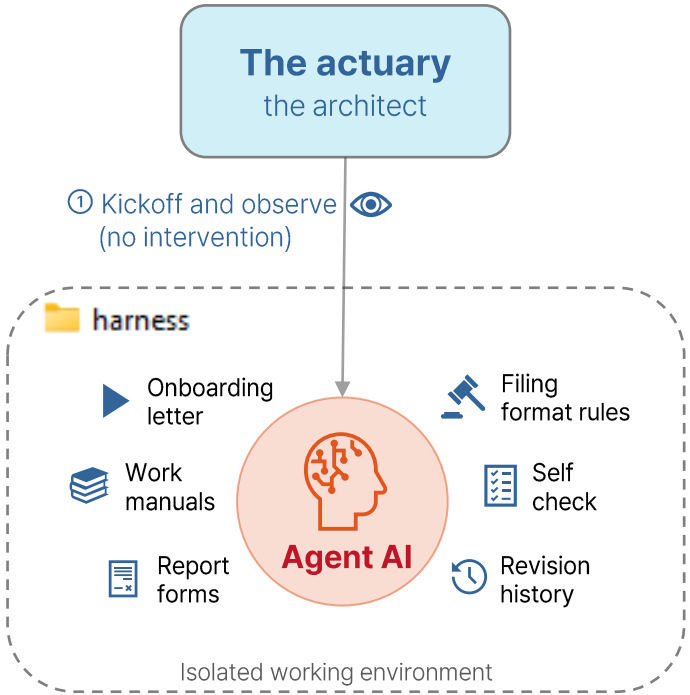
Munich RE

Agent AI in action… reading the kickoff prompt



The study cycle

Human ON the loop — not in it
(for this experiment)



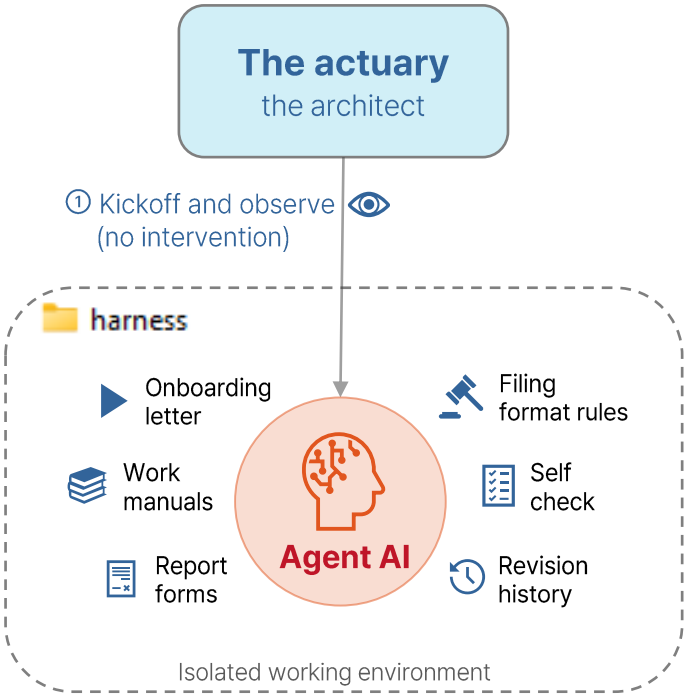
Munich RE

Agent AI in action... setting up its own tools — errors included

```
● I'll start by surveying the workspace - playbooks, schemas, templates, and the data

  Searched for 2 patterns, read 8 files (ctrl+o to expand)

● Bash(python3 -c "import pandas, pyarrow, numpy; print('pandas', pandas.__version__,
  └ pandas 2.3.3 | pyarrow 23.0.1 | numpy 2.3.5
    Traceback (most recent call last):
      File "<string>", line 1, in <module>
    … +12 lines (ctrl+o to expand)

● Bash(pip install duckdb 2>&1 | tail -2)
  └ note: If you believe this is a mistake, please contact your Python installation
    --break-system-packages.
    hint: See PEP 668 for the detailed specification.    ERROR

● Bash(pip install --user duckdb 2>&1 | tail -2 || pip install --break-system-packages
  └ note: If you believe this is a mistake, please contact your Python installation
    --break-system-packages.
    hint: See PEP 668 for the detailed specification.    ERROR

● Bash(pip install --break-system-packages duckdb 2>&1 | tail -3)
  └ ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 21.5/21.5 MB 14.2 MB/s eta 0:00:00
    Installing collected packages: duckdb
    Successfully installed duckdb-1.5.4                  SUCCESS

● DuckDB is installed. Setting up the task plan and workspace, then probing engine sem

✻ Ionizing… (1m 30s · ↓ 3.2k tokens)
  └ ☐ Profile submission data (schemas, keys, signs, cohorts)
    ☐ Design and apply data-quality treatments; build cleaned tables
    ☐ Write findings.json
    ☐ Mortality A/E analysis on treaty premium basis
    ☐ Lapse experience analysis
    … +3 pending
```

The study cycle

Human ON the loop — not in it
(for this experiment)



Agent AI in action... profiling one million policies

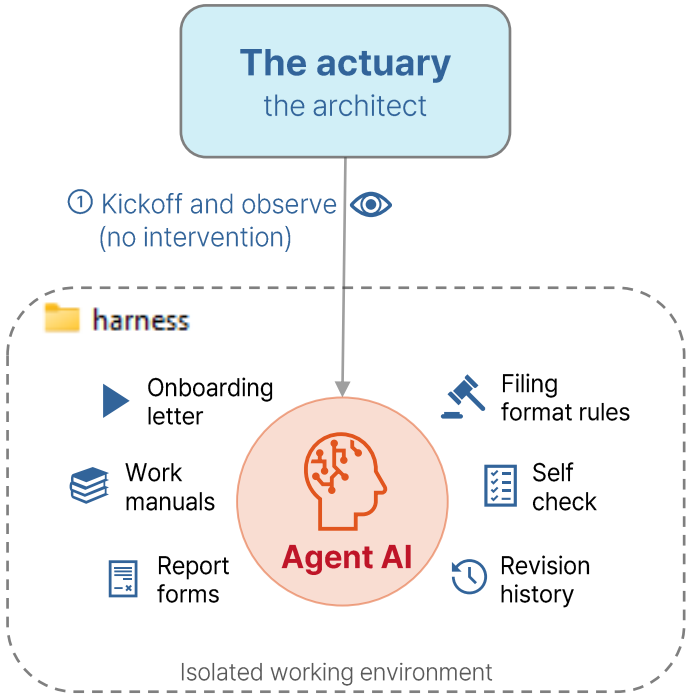
Munich RE

Experience study for Korea

```
print(con.execute("SELECT final_status, count(*) n, sum(CASE WHEN final_status_date IS NULL THEN
EOF)
=== issue_date range, by product ===
         product min(issue_date) max(issue_date)  count_star()
      WHOLE_LIFE      2019-01-01      2025-12-01        547129
TERM_5YR_RENEWABLE      2019-01-01      2025-12-01        452871

=== issue day-of-month distribution ===
 d  count_star()
 1       1000000

=== issue_age stats ===
 min(issue_age)  max(issue_age)  avg(issue_age)
             -1             999       43.126065
 issue_age  n
        -1 87
         0 80
       999 83

=== sum_assured stats ===
 min(sum_assured)  max(sum_assured)  median(sum_assured)  avg(sum_assured)
                0           1000000             120000.0         149385.51
(5,)

=== high_sum_assured vs sum_assured ===
 high_sum_assured  min(sum_assured)  max(sum_assured)  count_star()
             True            300000           1000000         87182
            False                 0            290000        912818

=== policy_id pattern ===
prefix  count_star()  min(length(policy_id))  max(length(policy_id))
     P       1000000                      14                      14

=== policy_id uniqueness ===
(1000000, 1000000)

=== final_status vs final_status_date nulls ===
     final_status      n  null_date
          INFORCE 625128   625128.0
            DEATH   2221        0.0
EXPIRY_NONRENEWAL  22808        0.0
            LAPSE 349843        0.0
```

The study cycle

Human ON the loop — not in it
(for this experiment)



Munich RE

Agent AI in action... catching the planted defects

```
print("\n=== claims not joining to pm ===")
print(con.execute("SELECT count(*) FROM cl c LEFT JOIN pm p USING(policy_id) WHERE p.policy_id IS NULL").
print("\n=== the 70 dup-claim policies: structure ===")
print(con.execute("""
WITH d AS (SELECT policy_id FROM cl GROUP BY 1 HAVING count(*)=2)
SELECT c.* , p.sum_assured
FROM cl c JOIN d USING(policy_id) JOIN pm p USING(policy_id)
ORDER BY c.policy_id, c.claim_id LIMIT 12
""").df().to_string(index=False))
EOF)
Running…
```

Claim Amount Error

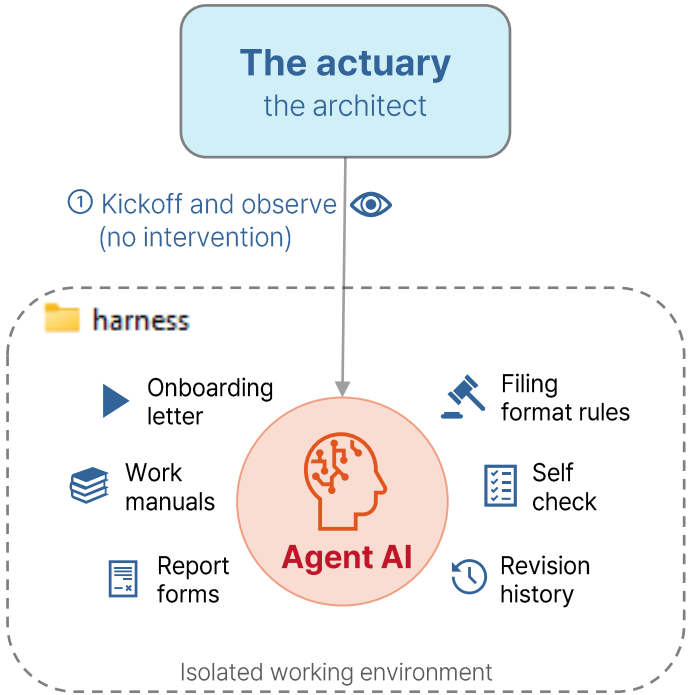
```
=== reins_claim vs sum_assured ===
claim_id      policy_id  reins_claim  sum_assured  ratio
C0001571 P2022090005128    180000000       180000 1000.0
C0001653 P2022100002798     60000000        60000 1000.0
C0001364 P2022100000416     80000000        80000 1000.0
C0001285 P2022110003944     70000000        70000 1000.0
C0001245 P2022110005730    100000000       100000 1000.0
C0001244 P2022100000411     60000000        60000 1000.0
C0001286 P2022110006234     90000000        90000 1000.0
C0001365 P2022110004397    190000000       190000 1000.0
C0001656 P2022110004548    260000000       260000 1000.0
C0001658 P2022120000067     50000000        50000 1000.0

=== ratio buckets ===
      b    n          amt
  equal 2015 2.845100e+08
  x1000  236 3.458000e+10
partial  140 1.053000e+07

=== claims not joining to pm ===
(50,)
```

Duplicate Claim Error

```
=== the 70 dup-claim policies: structure ===
  claim_id      policy_id date_of_death  yyyymm  attained_age  policy_year cause_of_death  reins_claim  s
C0000088-1 P2019010000859    2020-06-22  202006            67            2        DISEASE        78000
C0000088-2 P2019010000859    2020-06-22  202007            67            2        DISEASE        52000
C0001978-1 P2019010003546    2025-06-18  202506            66            7        DISEASE       186000
C0001978-2 P2019010003546    2025-06-18  202508            66            7        DISEASE       124000
C0000104-1 P2019020003006    2020-08-31  202008            62            2        DISEASE        24000
C0000104-2 P2019020003006    2020-08-31  202009            62            2        DISEASE        16000
C0000681-1 P2019020007423    2022-11-17  202211            61            4        DISEASE        36000
C0000681-2 P2019020007423    2022-11-17  202302            61            4        DISEASE        24000
C0000021-1 P2019030005162    2019-10-21  201910            70            1        DISEASE       162000
C0000021-2 P2019030005162    2019-10-21  201912            70            1        DISEASE       108000
```

The study cycle

Human ON the loop — not in it
(for this experiment)



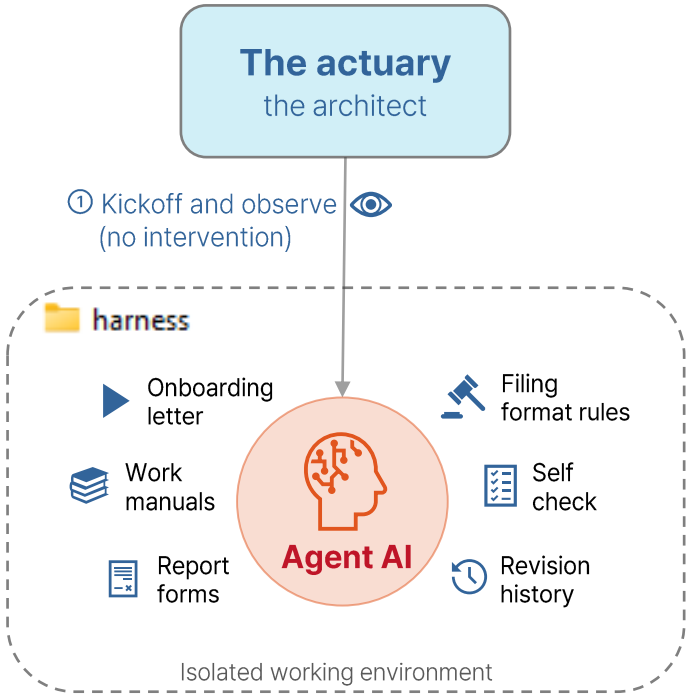
Munich RE

Agent AI in action… writing the full cleaning pipeline

```
    policy_id    event_type event_date
P2019070006069         ISSUE 2019-07-01
P2019070006069         LAPSE 2024-02-01
P2019070006069 REINSTATEMENT 2024-07-01
P2019070006069       RENEWAL 2024-07-01

Write(/home/agent/cedant_study/work/clean_pipeline.py)
└ Wrote 371 lines to /home/agent/cedant_study/work/clean_pipeline.py
     1 """Data-quality treatment pipeline. Reads ./data (read-only), writes
     2 ./outputs/cleaned/ plus a movement ledger (work/ledger.json).
     3
     4 Treatment order (dependencies annotated):
     5   P01 recode invalid yyyymm (X00 -> (X-1)12, X13 -> (X+1)01)
     6   P02 dedupe identical duplicate rows
     7   P03 net reversal triples (keep the member row equal to the group net)
     8   P04 /1000 scale repair on 2024 rows where nar == SA*1000 (needs master SA)
     9   P05 master repairs: sex sentinels, issue_age sentinels, SA=0 (needs resolver tables)
    10   P06 premium demographic repairs: sex flips, age flips, null sex/issue_age
    11   P07 sign-flip premiums (prm<0 -> |prm|)
    12   P08 zero premiums -> basis premium (needs P04, P06)
    13   P09 master status reclass: 100 grace deaths LAPSE -> DEATH
    14   P10 exclude post-death premium rows (needs P09 so grace-death policies keep window to death)
    15   P11 events: reconstruct missing REINSTATEMENT (271) at billing-resumption month
    16   P12 claims: exclude 50 orphans; /1000 repair 236; consolidate 70 split pairs
    17   P13 premium exposure reconstruction: 2 omitted issue cohorts, 500 scattered
    18       no-premium policies, grace months of the 100 grace deaths (needs P05, P09)
    19 """
    20 from db import con, DATA, WORK
    21 import json, os
    22
    23 L = {}  # ledger: step -> dict of counts
    24
    25 def log(step, **kw):
    26     L[step] = kw
    27     print(step, kw)
    28
    29 OUT = "/home/agent/cedant_study/outputs/cleaned"
    30 os.makedirs(OUT, exist_ok=True)
    31
    32 # ---------------------------------------------------------------- load raw
    33 con.execute(f"CREATE OR REPLACE TABLE c_prem AS SELECT policy_id, yyyymm, attained_age, policy_
       hly/*/*.parquet', hive_partitioning=true)")
    34 con.execute(f"CREATE OR REPLACE TABLE c_pm   AS SELECT * FROM read_parquet('{DATA}/policy_maste
    35 con.execute(f"CREATE OR REPLACE TABLE c_cl   AS SELECT * FROM read_parquet('{DATA}/reinsurance
```

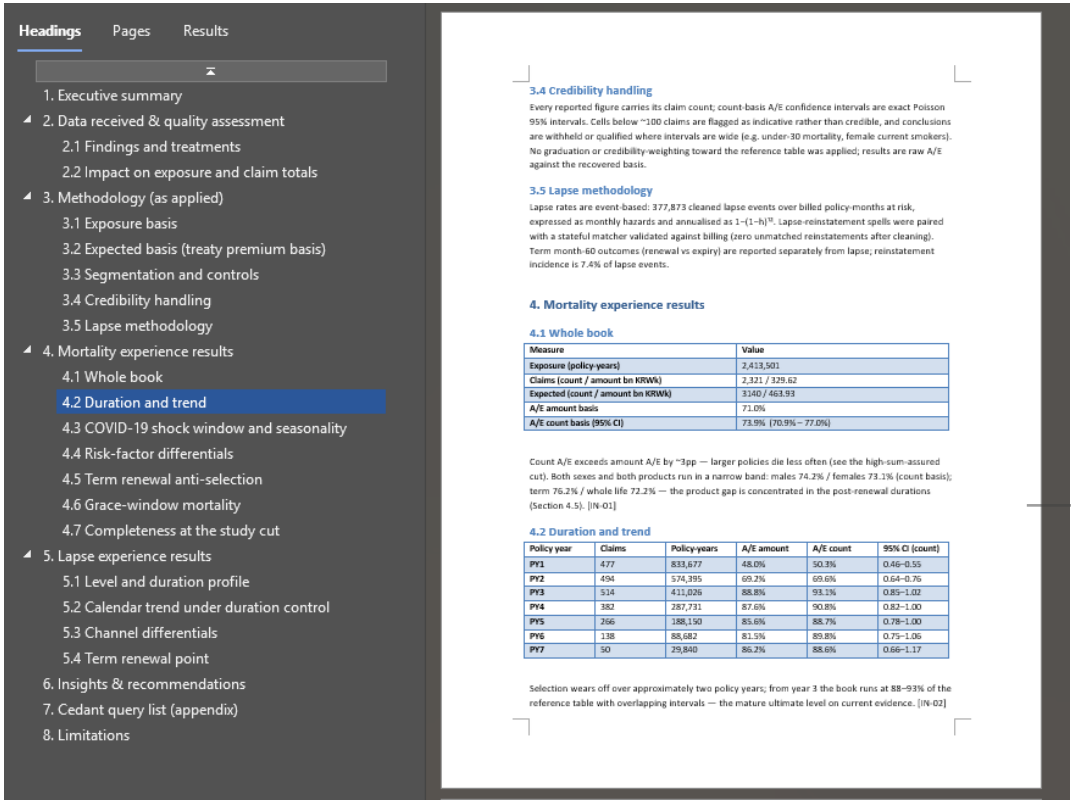
The study cycle

Human ON the loop — not in it
(for this experiment)



Munich RE

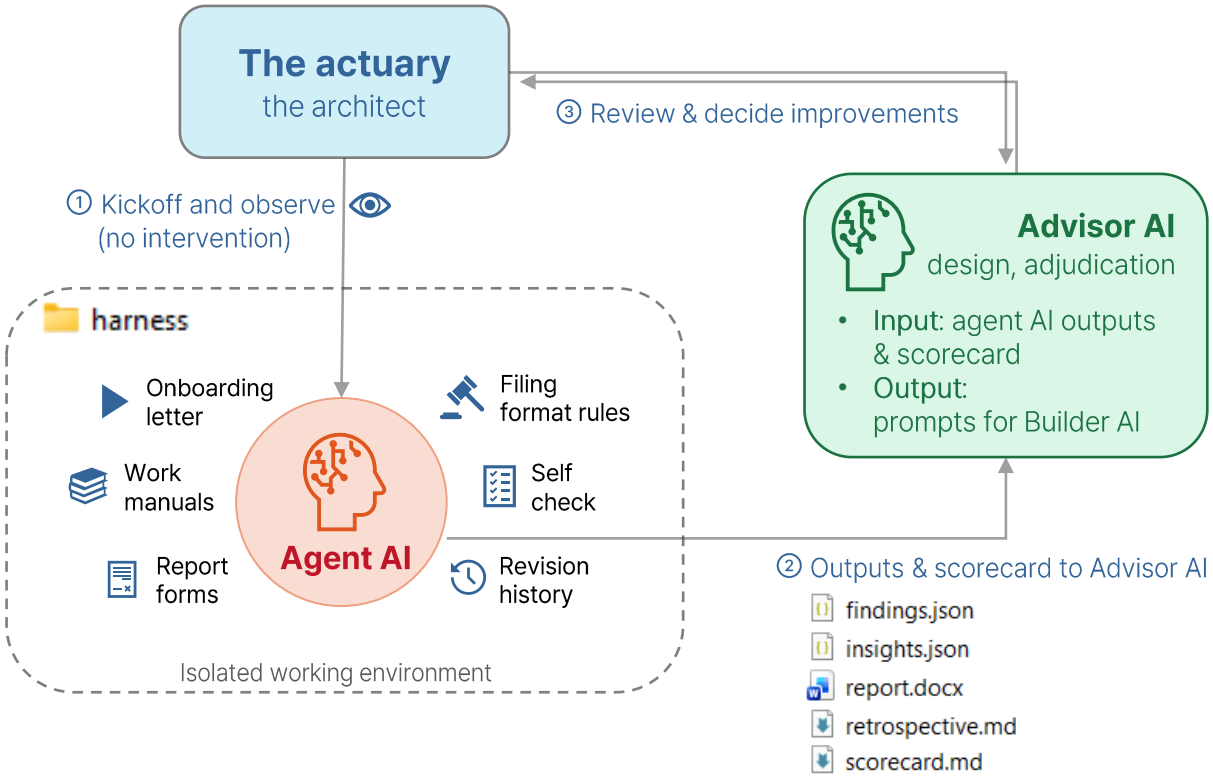
Study Report generated by Agent AI



The study cycle

Human ON the loop — not in it
(for this experiment)

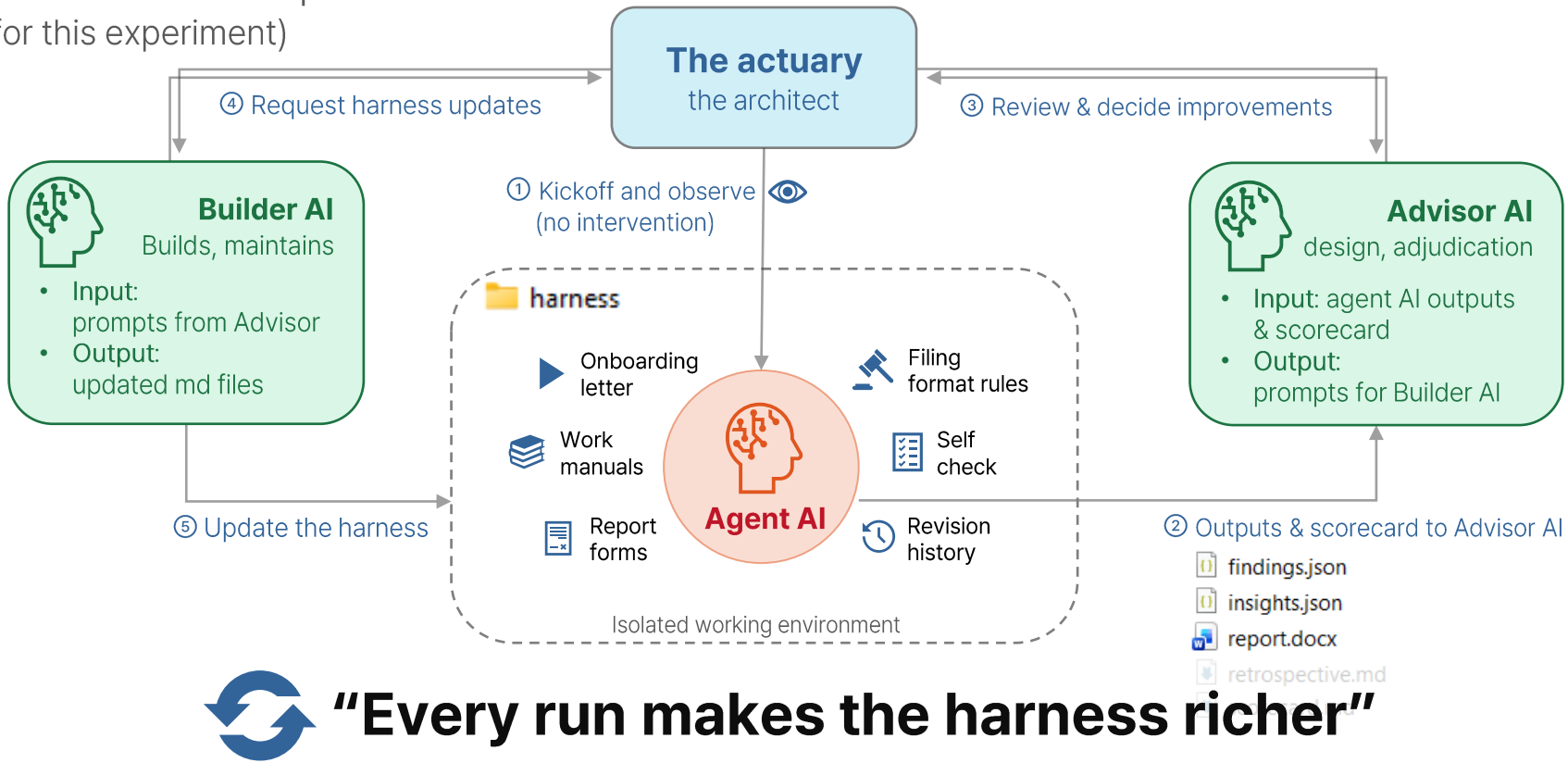
Munich RE



The study cycle

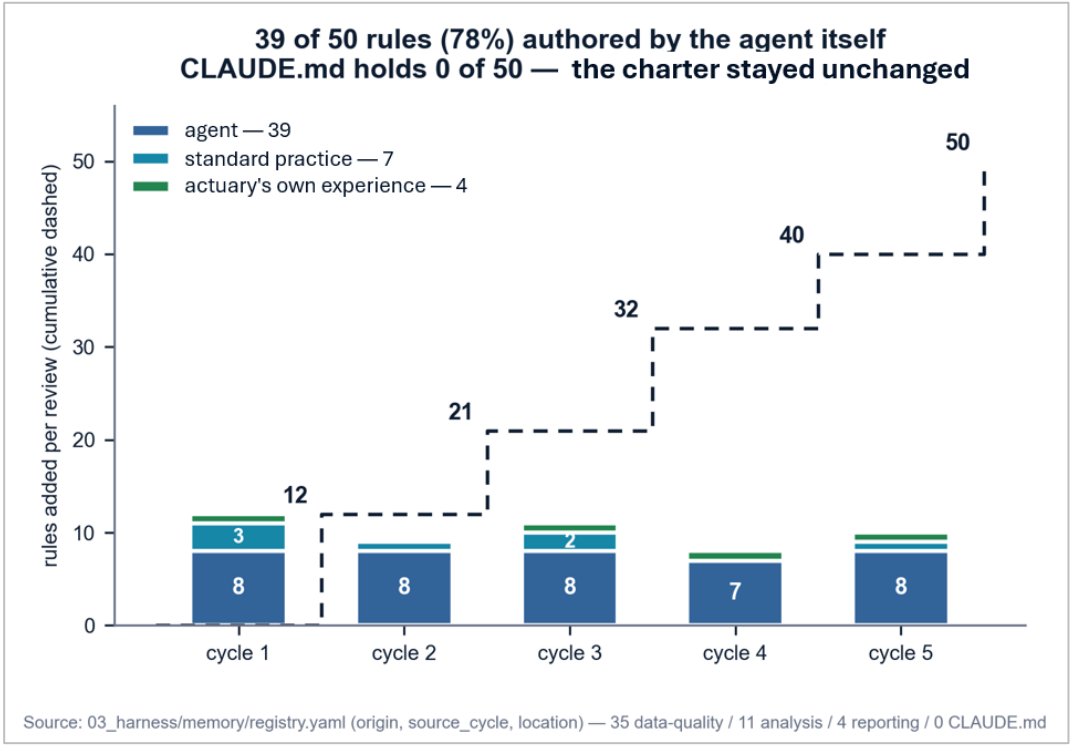
Human ON the loop — not in it
(for this experiment)

Munich RE



"Every run makes the harness richer"

The harness the agent wrote



The **agent** held the **pen**.
The **actuary** held the **gate**.

— and the gate is where the judgment lives.

Moments that surprised me

I designed the exam. The agent still surprised the examiner.

Munich RE

THE AGENT IMPROVED THE ANSWER KEY

“Unrepairable” — repaired.

100 policies planted with demographics destroyed beyond repair. The agent inverted the rate basis — all recovered, verifiably.

Our answer key said: exclude.

THE AGENT INVENTED A TECHNIQUE

February was hiding Omicron.

Winter seasonality and the Omicron wave, in the same month. The agent noticed — and separated the two.

Reviewed and approved.

THE AGENT SOLVED THE NEVER-SEEN

First encounter: 100%.

Two defect types it had never seen before. Both found and fixed on first sight — and nothing was flagged that wasn't a problem.

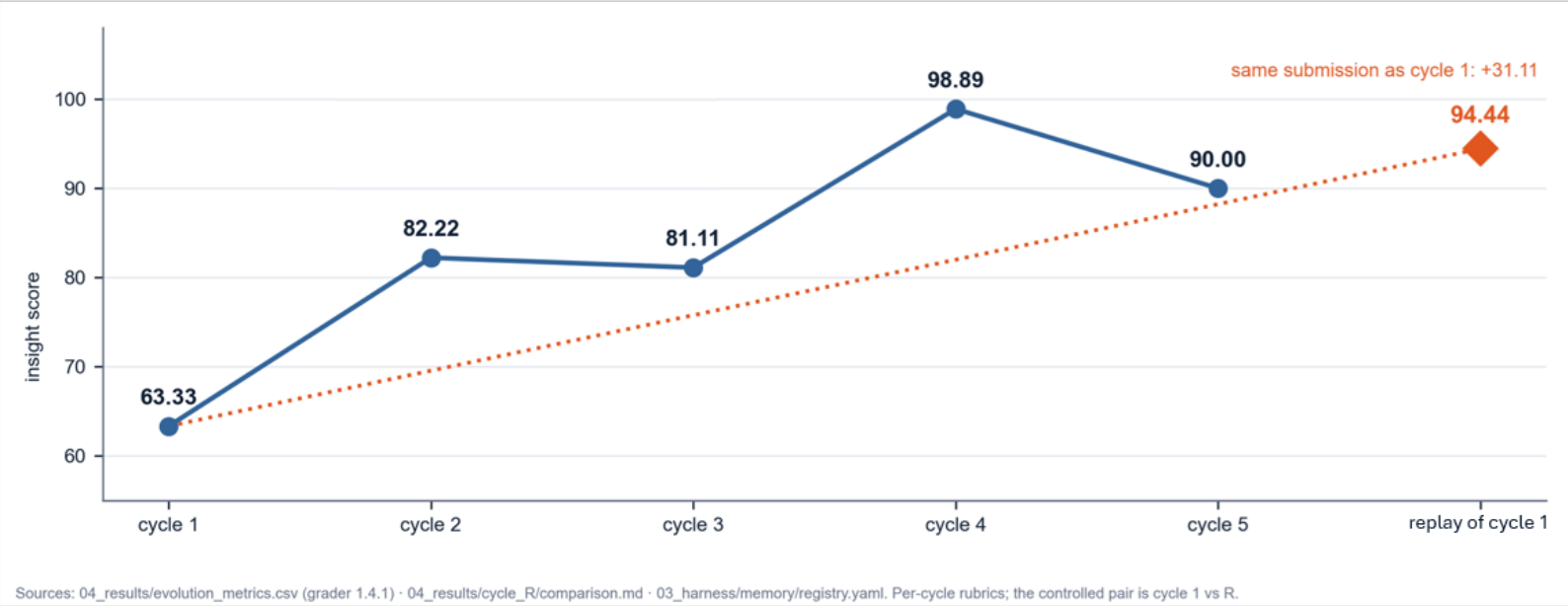
Then it proposed the fixes as new rules.

Scores across cycles

Judgment is where the harness made the difference

Munich RE

The wiggle is run-to-run noise. The climb is the harness



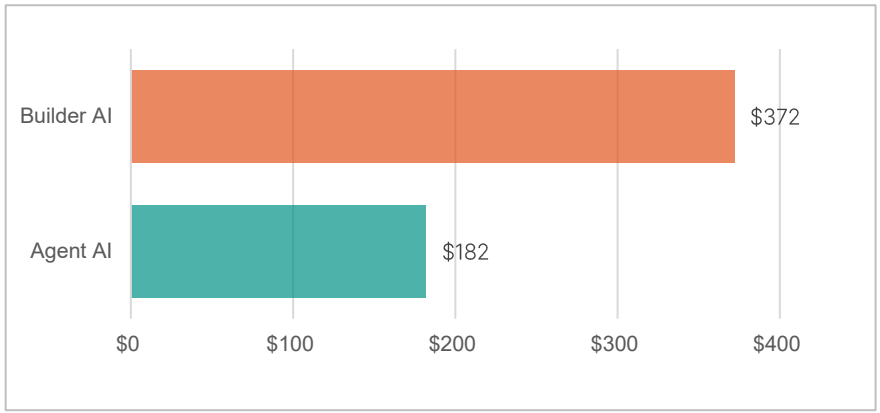
Sources: 04_results/evolution_metrics.csv (grader 1.4.1) · 04_results/cycle_R/comparison.md · 03_harness/memory/registry.yaml. Per-cycle rubrics; the controlled pair is cycle 1 vs R.

What did it cost?

The whole experiment, priced at API list rates

Munich RE

Costs vary widely with data complexity, model choice, and reasoning effort — treat this as one data point, not a benchmark.



- $554 — the entire experiment.
  Laboratory, grader, five review cycles, six study runs, close-out.
- $24–$35 per study run.
  About 90 minutes of model-active time each.
- Builder ≈ 2× Agent.
  The investment sits in the environment — and the environment is reusable.

- Claude Fable 5, xhigh reasoning effort, via Claude Code
- All figures are API-list-price equivalents computed from session transcripts (runs were subscription-included)

From Analyst to Harness Architect

Munich RE

The expertise didn't leave the profession — it moved into the environment

▪ Data Quality Judgment	→	Written Playbooks: how to find and fix bad data
▪ Study Methodology	→	Same principles, now written Analysis Rules
▪ Knowing what matters	→	Rules for what to chase — and what to let go
▪ Peer Review	→	Review gate: no rule enters without the actuary's approval
▪ Professional Accountability	→	The agent conducts the study — the actuary signs it

Can an AI Agent Conduct an Experience Study?

Yes —

when an actuary builds the harness.

Slides & more

19 August 2026
HyunSu Kim, Munich Re

Munich RE

The views expressed are my own and do not necessarily represent those of Munich Re

Image source: AI-generated image created by HyunSu Kim

Appendix

Building the Test Environment

Synthetic data with known defects, ~31M records

Munich RE

Step 1: Generate a clean baseline

1 Million synthetic policies, Korean life portfolio

- Whole life and 5-yr renewable term, issued 2019–2025, monthly records
- Mortality: 10th KMET base with multiplicative factors (selection, improvement, COVID-19, smoker, UW class, channel)
- Lognormal frailty drives endogenous renewal anti-selection
- Monthly simulation: each policy faces death or lapse, never both

Step 2: Inject realistic defects

A "cedant submission" with 13+2 planted defect types

- Mechanical: duplicates, invalid codes, orphan claims
- Actuarial: premium after death, cohort gaps, split claims
- Judgment: code scheme changes, missing events, mid-history corrections
- Every defect logged with its correct resolution: detection and repair scored automatically

10th KMET: Korean Mortality Experience Table

Appendix

Building the Test Environment

Synthetic data with known defects, ~31M records

Munich RE

Policy Master (1M)

	A	B	C	D	E	F	G	H	I	J	K	L
1	policy_id	product	issue_date	issue_age	sex	smoking_status	underwriting_class	channel	sum_assured	high_sum_assured	final_status	final_status_date
2	P2019010000001	TERM_5YR_	1/1/2019	34	M	NEVER	PREFERRE	GA	240000	FALSE	LAPSE	12/1/2019
3	P2019010000002	WHOLE_LIF	1/1/2019	38	F	NEVER	STANDARI	GA	30000	FALSE	INFORCE	
4	P2019010000003	TERM_5YR_	1/1/2019	58	M	CURRENT	PREFERRE	GA	490000	TRUE	LAPSE	5/1/2019
5	P2019010000004	WHOLE_LIF	1/1/2019	38	M	FORMER	STANDARI	GA	50000	FALSE	LAPSE	10/1/2023
6	P2019010000005	WHOLE_LIF	1/1/2019	26	M	FORMER	PREFERRE	GA	190000	FALSE	LAPSE	9/1/2020
7	P2019010000006	TERM_5YR_	1/1/2019	50	M	FORMER	STANDARI	GA	240000	FALSE	INFORCE	
8	P2019010000007	WHOLE_LIF	1/1/2019	55	F	NEVER	STANDARI	GA	100000	FALSE	LAPSE	10/1/2021
9	P2019010000008	TERM_5YR_	1/1/2019	68	M	CURRENT	STANDARI	GA	70000	FALSE	DEATH	6/26/2019
10	P2019010000009	WHOLE_LIF	1/1/2019	56	F	NEVER	STANDARI	GA	60000	FALSE	LAPSE	1/1/2019
11	P2019010000010	TERM_5YR_	1/1/2019	44	M	FORMER	STANDARI	GA	210000	FALSE	LAPSE	4/1/2019

Policy Events (~1.4M)

	A	B	C
1	policy_id	event_type	event_date
2	P2019010(	ISSUE	1/1/2019
3	P2019010(	LAPSE	12/1/2019
4	P2019010(	ISSUE	1/1/2019
5	P2019010(	ISSUE	1/1/2019
6	P2019010(	LAPSE	5/1/2019
7	P2019010(	ISSUE	1/1/2019
8	P2019010(	LAPSE	7/1/2021
9	P2019010(	REINSTATEMENT	8/1/2021
10	P2019010(	LAPSE	10/1/2023

Reinsurance Premium (~29M)

	A	B	C	D	E	F
1	policy_id	yyyymm	attained_age	policy_year	ceded_nar	reins_premium
2	P2019010(	201901	34	1	240000	10.26
3	P2019010(	201901	38	1	30000	0.99
4	P2019010(	201901	58	1	490000	115.59
5	P2019010(	201901	38	1	50000	2.5
6	P2019010(	201901	26	1	190000	6.35
7	P2019010(	201901	50	1	240000	28.2
8	P2019010(	201901	55	1	100000	8.75
9	P2019010(	201901	68	1	70000	44.33
10	P2019010(	201901	56	1	60000	5.55

Reinsurance Claim (~2.4K)

	A	B	C	D	E	F	G	H
1	claim_id	policy_id	date_of_death	yyyymm	attained_age	policy_year	cause_of_death	reins_claim
2	C0000001	P2019010(	6/26/2019	201906	68	1	DISEASE	70000
3	C0000002	P2019070(	11/21/2020	202011	59	2	DISEASE	240000
4	C0000003	P2021040(	4/16/2021	202104	47	1	DISEASE	510000
5	C0000004	P2019070(	3/11/2022	202203	72	3	COVID19	440000
6	C0000005	P2021020(	5/17/2022	202205	68	2	DISEASE	170000
7	C0000006	P2020090(	7/5/2022	202207	42	2	ACCIDENT	40000
8	C0000007	P2020090(	8/3/2022	202208	52	2	COVID19	40000
9	C0000008	P2020100(	9/17/2022	202209	62	2	DISEASE	150000
10	C0000009	P2022060(	9/25/2022	202209	53	1	DISEASE	280000