



Image source: AI-generated image created by HyunSu Kim

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



HyunSu Kim, ASA (just call me HS)

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



- 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



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



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

LLM

The Performer



The core capability that produces the output.

Harness Engineering

A term from the AI industry, early 2026

LLM

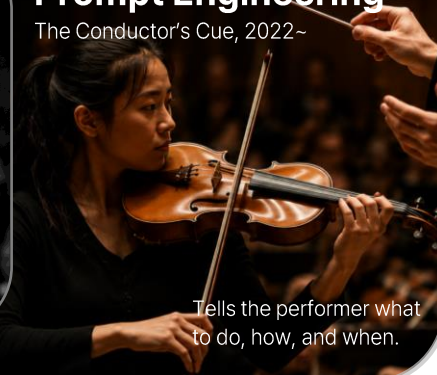
The Performer



The core capability that produces the output.

Prompt Engineering

The Conductor's Cue, 2022~



Tells the performer what to do, how, and when.

Harness Engineering

A term from the AI industry, early 2026

LLM

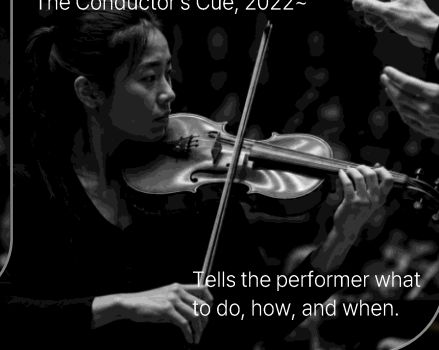
The Performer



The core capability that produces the output.

Prompt Engineering

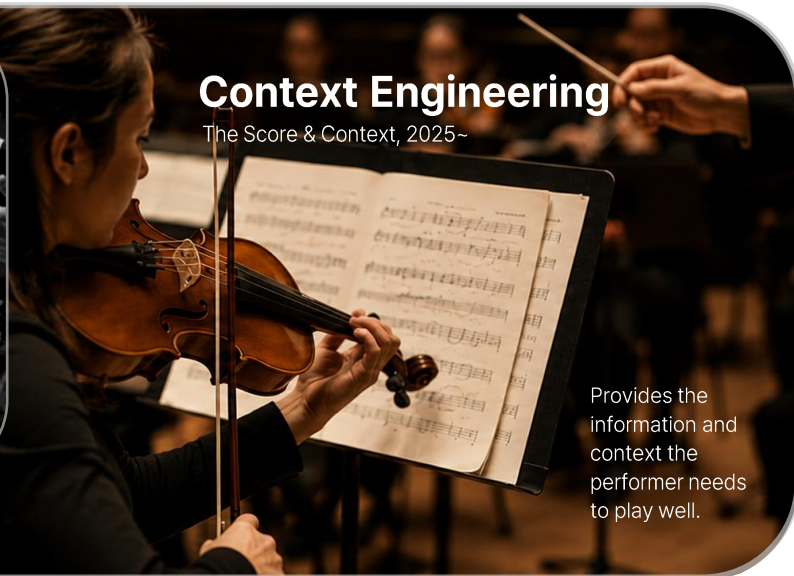
The Conductor's Cue, 2022~



Tells the performer what to do, how, and when.

Context Engineering

The Score & Context, 2025~



Provides the information and context the performer needs to play well.

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



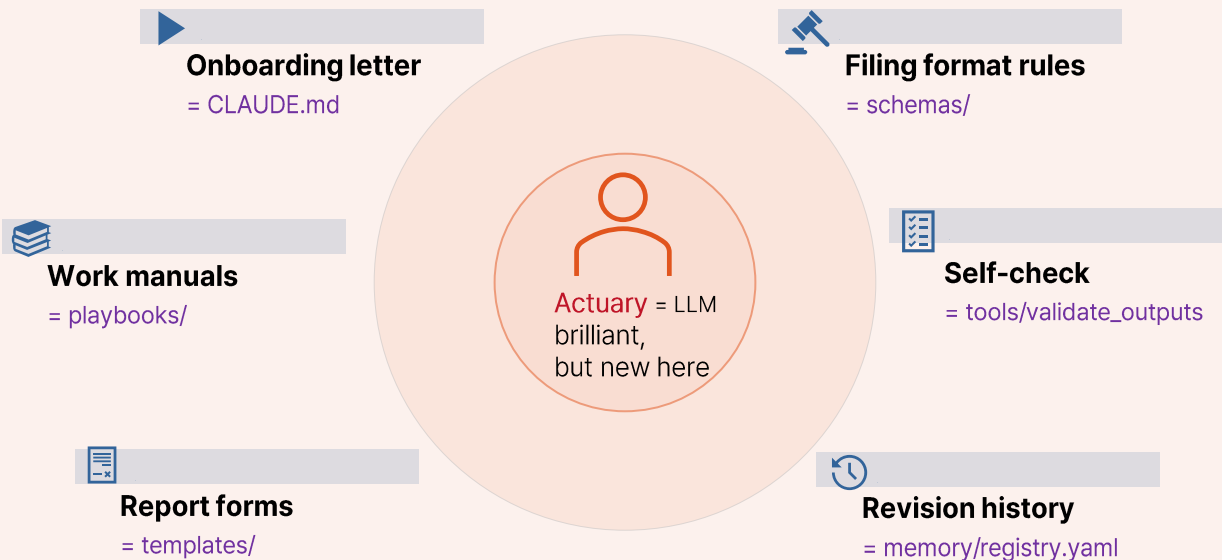
Guardrails & Recovery

Handle errors, adapt, and continue

Harness Engineering

A term from the AI industry, early 2026

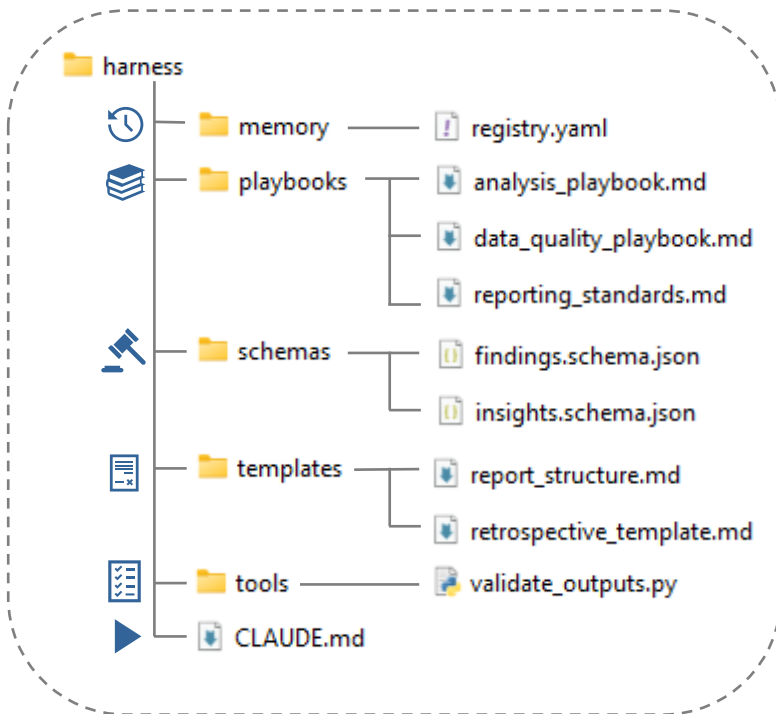
If the model is a talented new analyst, the **Harness** is everything the firm builds around them.



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

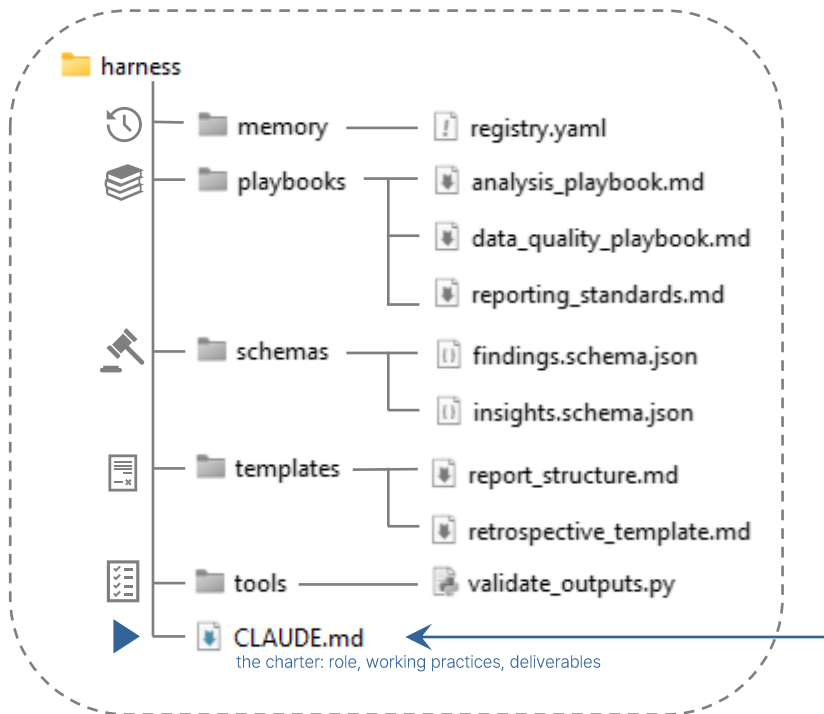
Inside the Harness

The harness is just text files



Inside the Harness

The harness is just text files



The charter: role, working practices, deliverables

```
CLAUDE.md

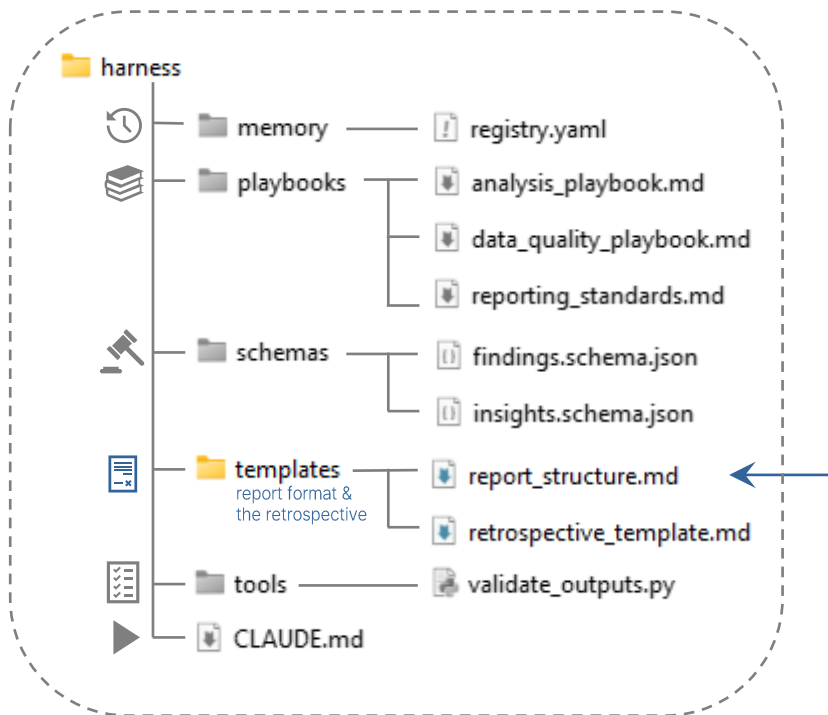
# Experience Study Workspace

## 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,
```

Inside the Harness

The harness is just text files



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

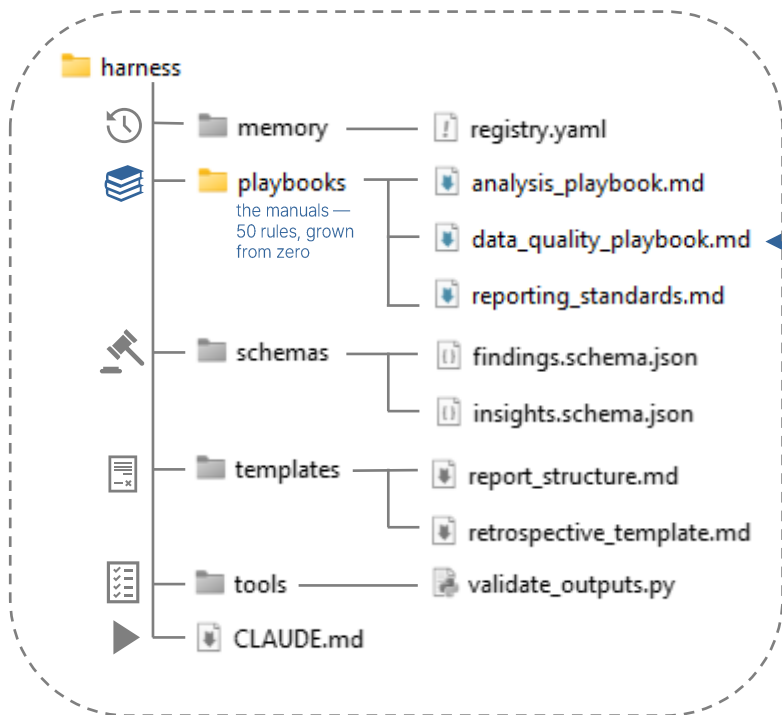
1. **Executive summary**
2. **Data received & quality assessment** – the issues you found, the treatment you applied to each, and the impact on exposure and claim totals (before vs after your cleaning).
3. **Methodology (as applied)** – describe what you actually did: exposure basis, expected basis, 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

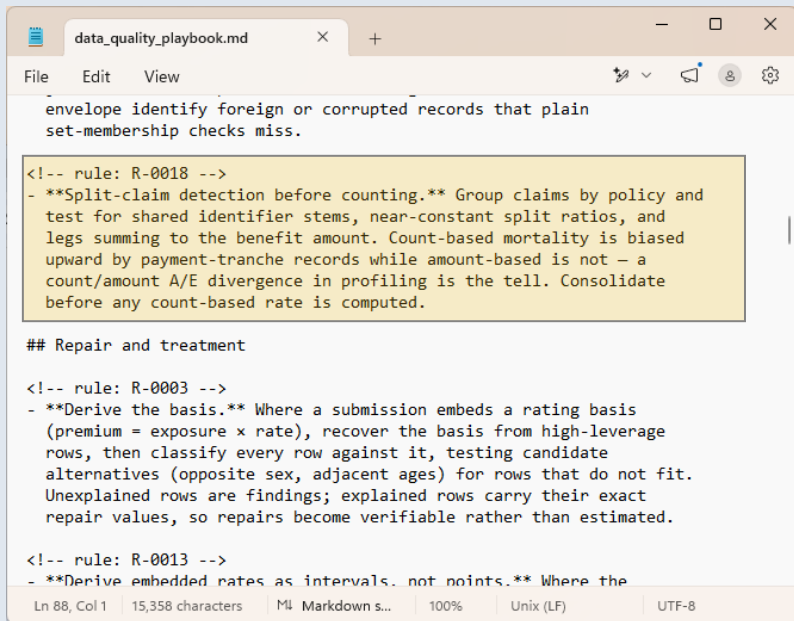
Ln 1, Col 1 | 1,265 characters | M4 Markdown s... | 100% | Unix (LF) | UTF-8
```

Inside the Harness

The harness is just text files



The work manuals: 50 rules, grown from zero



```
data_quality_playbook.md
File Edit View
envelope identify foreign or corrupted records that plain
set-membership checks miss.

<!-- rule: R-0018 -->
- **Split-claim detection before counting.** Group claims by policy and
test for shared identifier stems, near-constant split ratios, and
legs summing to the benefit amount. Count-based mortality is biased
upward by payment-tranche records while amount-based is not – a
count/amount A/E divergence in profiling is the tell. Consolidate
before any count-based rate is computed.

## Repair and treatment

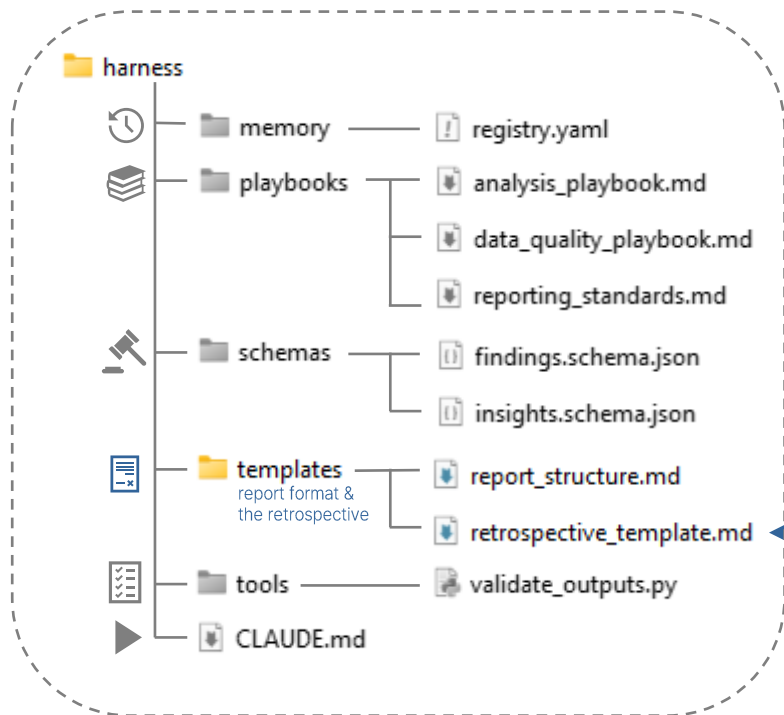
<!-- rule: R-0003 -->
- **Derive the basis.** Where a submission embeds a rating basis
(premium = exposure x rate), recover the basis from high-leverage
rows, then classify every row against it, testing candidate
alternatives (opposite sex, adjacent ages) for rows that do not fit.
Unexplained rows are findings; explained rows carry their exact
repair values, so repairs become verifiable rather than estimated.

<!-- rule: R-0013 -->
- **Derive embedded rates as intervals. not points.** Where the
```

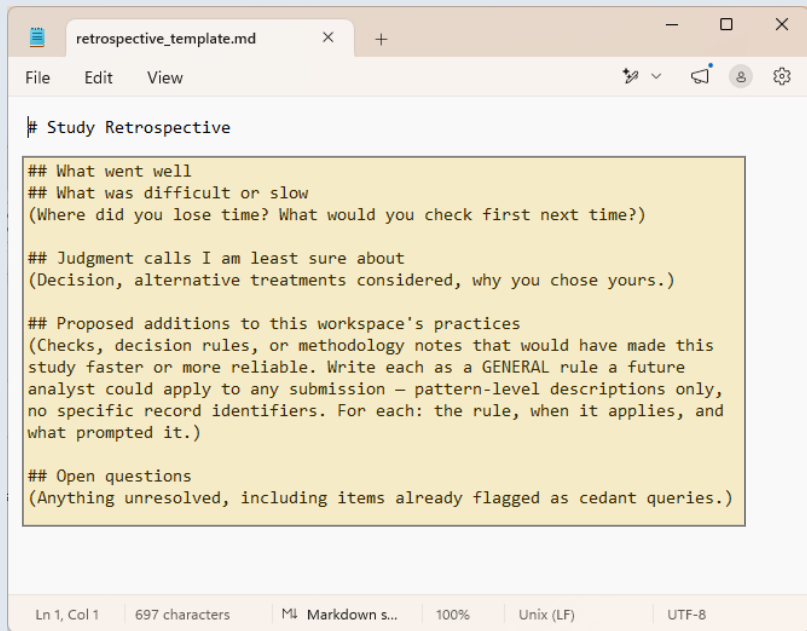
The screenshot shows a text editor window titled 'data_quality_playbook.md'. The editor contains text describing rules for data quality, including a rule for split-claim detection and a rule for deriving the basis. The text is formatted with markdown, including comments and bold text. The editor also shows a status bar at the bottom with information like 'Ln 88, Col 1', '15,358 characters', and 'ML Markdown s... 100% Unix (LF) UTF-8'.

Inside the Harness

The harness is just text files



Retrospective template: where the loop feeds itself



```
retrospective_template.md
File Edit View
# Study Retrospective

## What went well
## What was difficult or slow
(Where did you lose time? What would you check first next time?)

## Judgment calls I am least sure about
(Decision, alternative treatments considered, why you chose yours.)

## Proposed additions to this workspace's practices
(Checks, decision rules, or methodology notes that would have made this study faster or more reliable. Write each as a GENERAL rule a future analyst could apply to any submission – pattern-level descriptions only, no specific record identifiers. For each: the rule, when it applies, and what prompted it.)

## Open questions
(Anything unresolved, including items already flagged as cedant queries.)

Ln 1, Col 1 697 characters M4 Markdown s... 100% Unix (LF) UTF-8
```

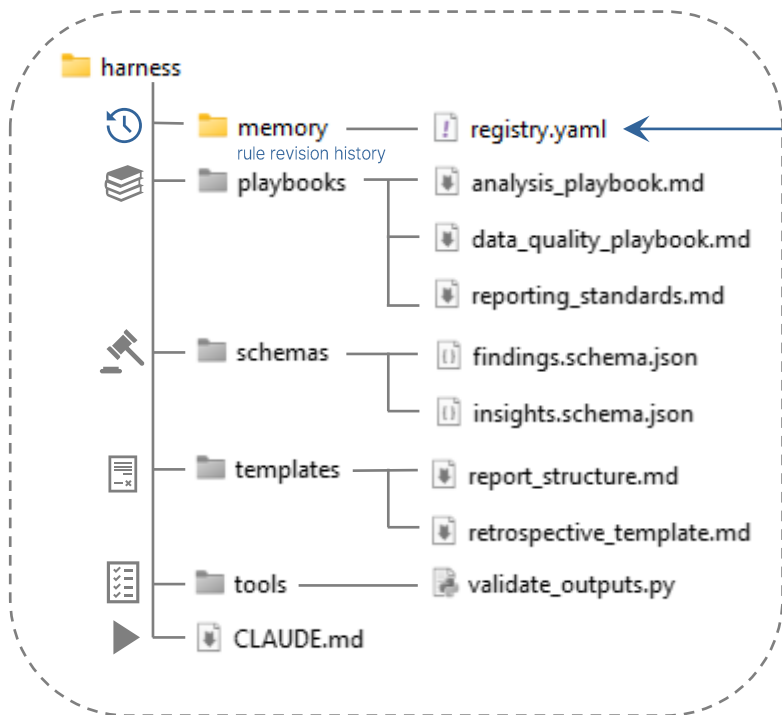
The screenshot shows a code editor window titled 'retrospective_template.md'. The editor contains a Markdown document with the following content:

- # Study Retrospective**
- ## What went well**
- ## What was difficult or slow**
(Where did you lose time? What would you check first next time?)
- ## Judgment calls I am least sure about**
(Decision, alternative treatments considered, why you chose yours.)
- ## Proposed additions to this workspace's practices**
(Checks, decision rules, or methodology notes that would have made this study faster or more reliable. Write each as a GENERAL rule a future analyst could apply to any submission – pattern-level descriptions only, no specific record identifiers. For each: the rule, when it applies, and what prompted it.)
- ## Open questions**
(Anything unresolved, including items already flagged as cedant queries.)

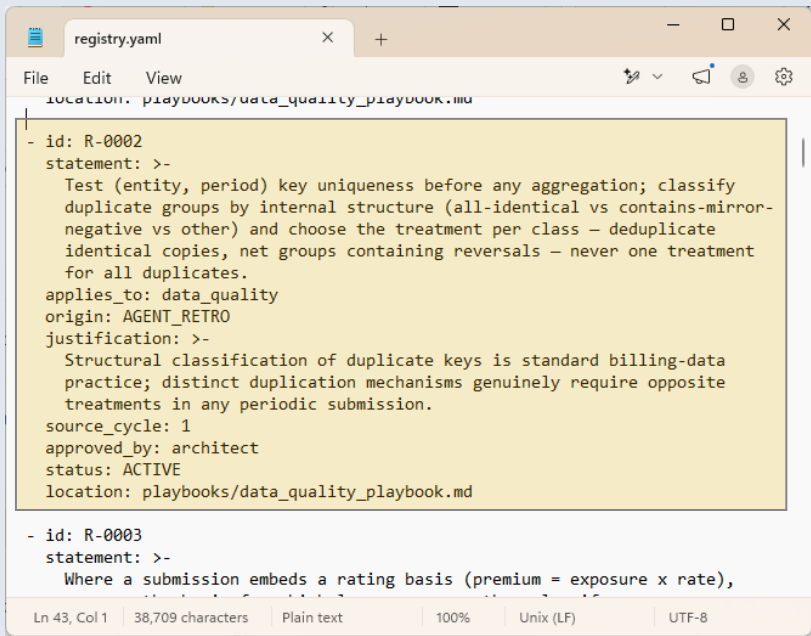
The status bar at the bottom indicates the cursor is at Line 1, Column 1, the file is 697 characters, it's a Markdown file (M4), the encoding is UTF-8, and the line endings are Unix (LF).

Inside the Harness

The harness is just text files



The memory: every rule's origin on record



```
registry.yaml
File Edit View
location: playbooks/data_quality_playbook.md

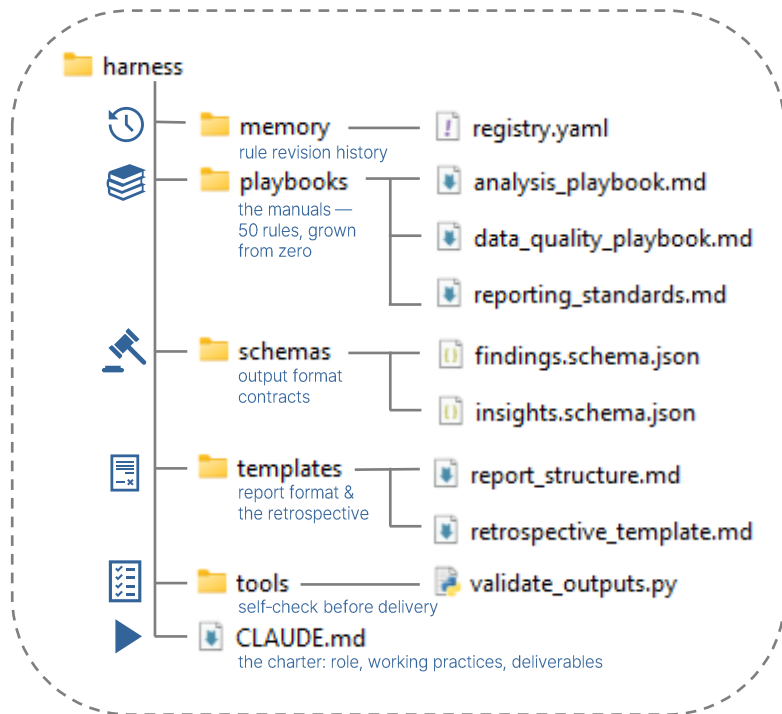
- id: R-0002
  statement: >-
    Test (entity, period) key uniqueness before any aggregation; classify
    duplicate groups by internal structure (all-identical vs contains-mirror-
    negative vs other) and choose the treatment per class – deduplicate
    identical copies, net groups containing reversals – never one treatment
    for all duplicates.
  applies_to: data_quality
  origin: AGENT_RETRO
  justification: >-
    Structural classification of duplicate keys is standard billing-data
    practice; distinct duplication mechanisms genuinely require opposite
    treatments in any periodic submission.
  source_cycle: 1
  approved_by: architect
  status: ACTIVE
  location: playbooks/data_quality_playbook.md

- id: R-0003
  statement: >-
    Where a submission embeds a rating basis (premium = exposure x rate),
```

The screenshot shows a text editor window titled 'registry.yaml'. The file contains a list of rules, each with a unique ID, a statement, and various metadata fields. The first rule, R-0002, describes a test for key uniqueness and its application to data quality. The second rule, R-0003, describes a rating basis calculation. The editor interface includes a menu bar (File, Edit, View), a status bar at the bottom showing line and column numbers (Ln 43, Col 1), character count (38,709 characters), encoding (UTF-8), and line ending (Unix (LF)).

Inside the Harness

The harness is just text files

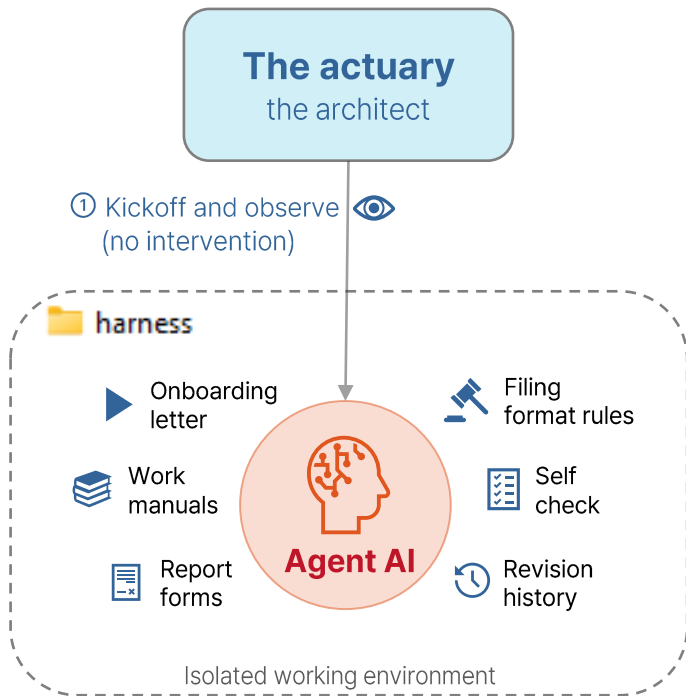


No code. No prompt tricks.

The same documentation
we'd give a new actuary.

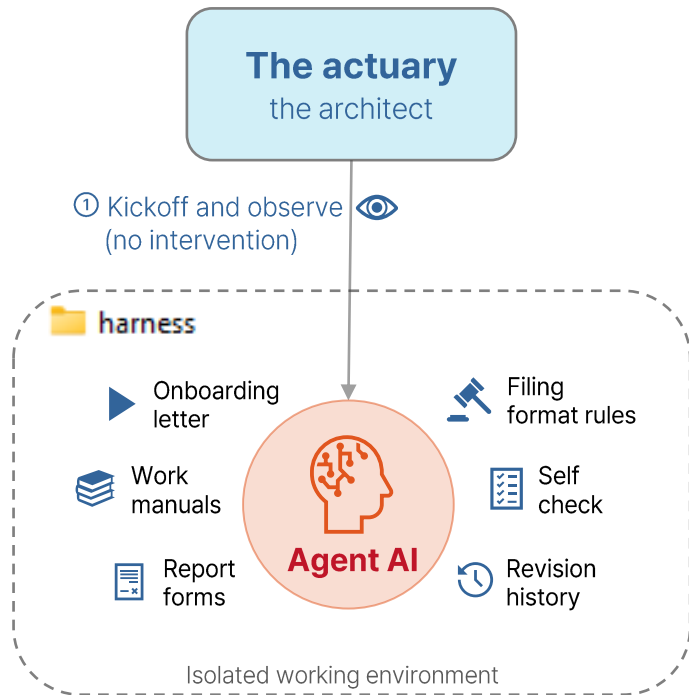
The study cycle

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



The study cycle

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



Agent AI in action... reading the kickoff prompt

```
C:\work\mulberry\02_evaluation\src>deploy_cycle_wsl.sh 1

C:\work\mulberry\02_evaluation\src>run_agent_cycle_wsl.cmd
Claude Code v2.1.200
Fable 5 with xhigh effort · Claude Max
~/cedant_study
```

Kickoff Prompt

```
A cedant submission has arrived in ./data/ - full-period bordereaux for
our 100% quota-share YRT treaty on their Korean whole life and 5-year
renewable term book, January 2019 through December 2025.

Conduct the full experience study: assess and treat data quality, then
analyze mortality experience (actual vs expected on the treaty premium
basis) and lapse experience, across the risk dimensions present in the
data. Produce all deliverables specified in CLAUDE.md.

Work autonomously end to end; there is no one available to answer
questions mid-study, so record anything you would ask the cedant as
cedant-query items. You have up to 4 hours. Begin.
```

- I'll start by surveying the workspace - data files, schemas, templates, and too

Listed 1 directory (ctrl+o to expand)

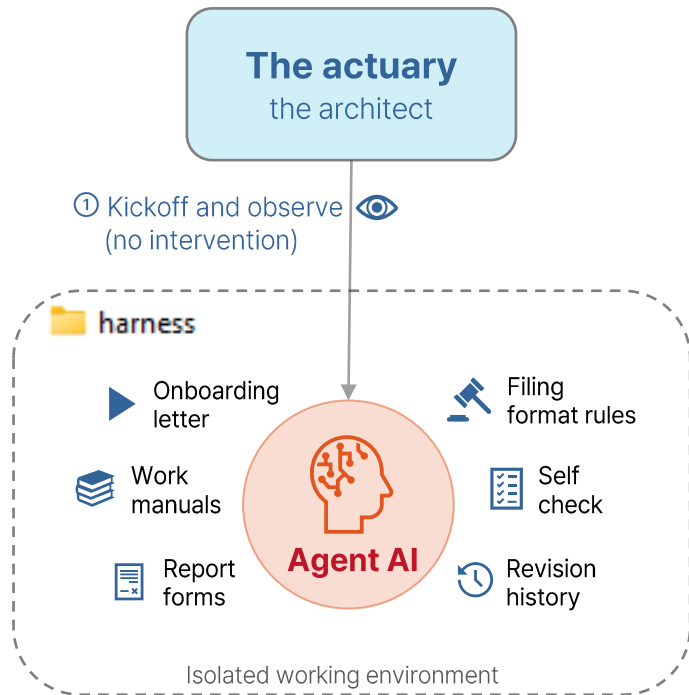
- Bash(python -c "import pandas, pyarrow, numpy; print(pandas.__version__, pyarrow

L Waiting...

Bash command

The study cycle

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



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__,
  L 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)
  L note: If you believe this is a mistake, please contact your Python installation o
  --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
  L note: If you believe this is a mistake, please contact your Python installation o
  --break-system-packages.
  hint: See PEP 668 for the detailed specification. ERROR

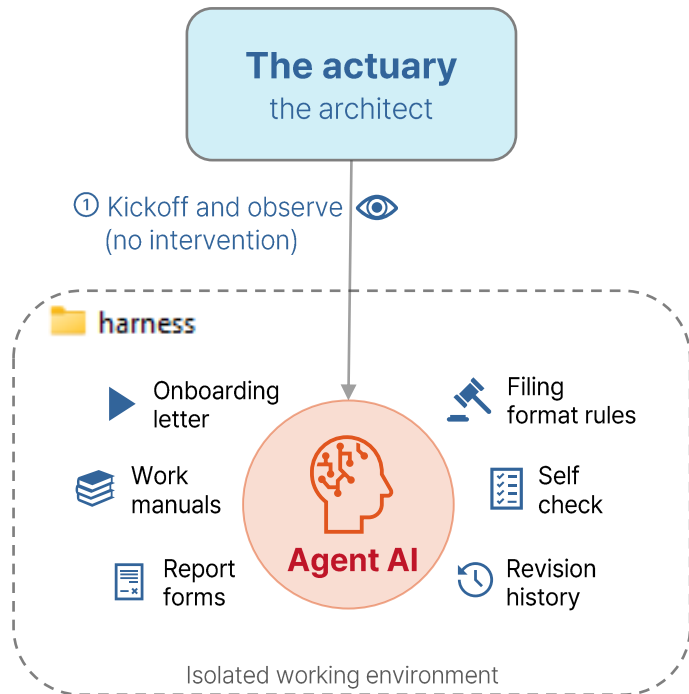
• Bash(pip install --break-system-packages duckdb 2>&1 | tail -3)
  L _____ 21.5/21.5 MB 14.2 MB/s eta 0:00:00
  Installing collected packages: duckdb
  Successfully installed duckdb-1.5.4 SUCCESS

• DuckDB is installed. Setting up the task plan and workspace, then probing engine sen

* Ionizing... (1m 30s · ↓ 3.2k tokens)
  L □ 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

```
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)
(5,)
0 1000000 120000.0 149385.51

=== 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

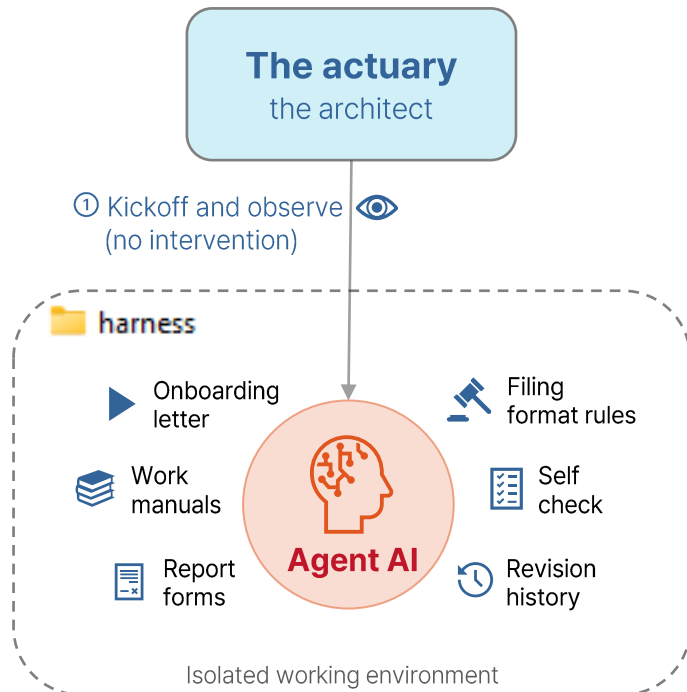
=== policy_id pattern ===
prefix count_star() min(length(policy_id)) max(length(policy_id))
p 1000000 14 14

=== 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)



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)
```

Claim Amount Error

```
Running...
=== 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
```

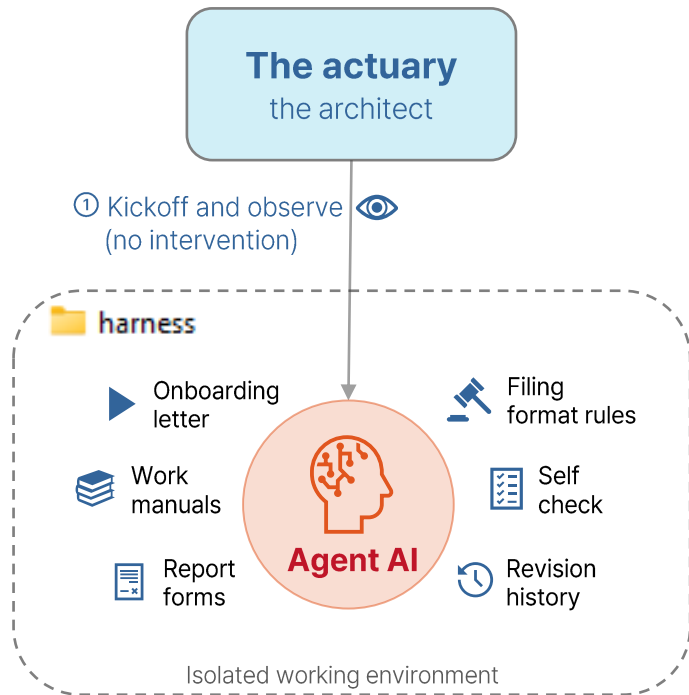
```
=== claims not joining to pm ===
(50,)
```

Duplicate Claim Error

```
=== the 70 dup-claim policies: structure ===
claim_id  policy_id  date_of_death  yyyyymm  attained_age  policy_year  cause_of_death  reins_claim
C0000088-1  P201901000859  2020-06-22  202006  67  2  DISEASE  78000
C0000088-2  P201901000859  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)



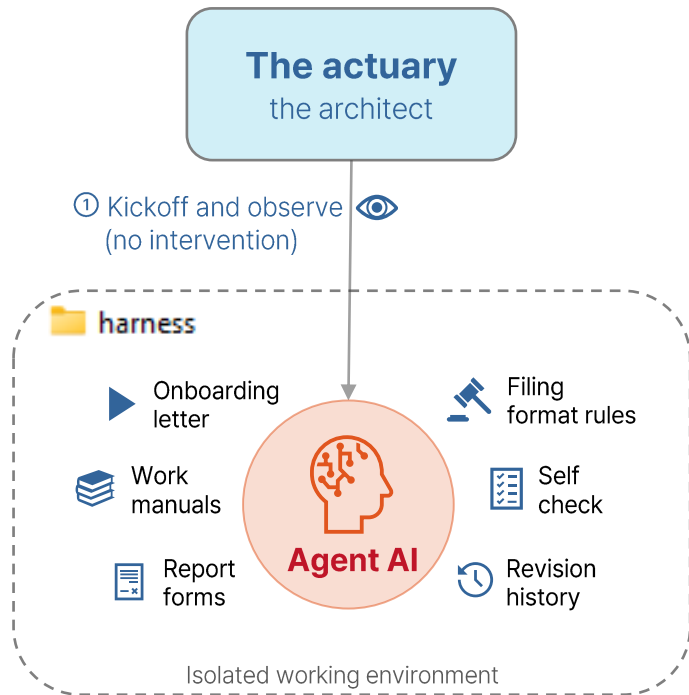
Agent AI in action... writing the full cleaning pipeline

```
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)
L 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 yyyy-mm (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, yyyy-mm, attained_age, policy_
34 hly/*.*.parquet', hive_partitioning=true)")
35 con.execute(f"CREATE OR REPLACE TABLE c_pm AS SELECT * FROM read_parquet('{DATA}/policy_maste
36 AS SELECT * FROM read_parquet('{DATA}/reinsurance
```


The study cycle

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



Study Report generated by Agent AI

Headings

Pages

Results

1. Executive summary

2. Data received & quality assessment

2.1 Findings and treatments

2.2 Impact on exposure and claim totals

3. Methodology (as applied)

3.1 Exposure basis

3.2 Expected basis (treaty premium basis)

3.3 Segmentation and controls

3.4 Credibility handling

3.5 Lapse methodology

4. Mortality experience results

4.1 Whole book

4.2 Duration and trend

4.3 COVID-19 shock window and seasonality

4.4 Risk-factor differentials

4.5 Term renewal anti-selection

4.6 Grace-window mortality

4.7 Completeness at the study cut

5. Lapse experience results

5.1 Level and duration profile

5.2 Calendar trend under duration control

5.3 Channel differentials

5.4 Term renewal point

6. Insights & recommendations

7. Cedant query list (appendix)

8. Limitations

3.4 Credibility handling

Every reported figure carries its claim count; count-basis A/E confidence intervals are exact Poisson 95% intervals. Cells below ~100 claims are flagged as indicative rather than credible, and conclusions are withheld or qualified where intervals are wide (e.g. under-30 mortality, female current smokers). No graduation or credibility-weighting toward the reference table was applied; results are raw A/E against the recovered basis.

3.5 Lapse methodology

Lapse rates are event-based: 377,873 cleaned lapse events over billed policy-months at risk, expressed as monthly hazards and annualised as $1-(1-h)^{12}$. Lapse-reinstatement spells were paired with a stateful matcher validated against billing (zero unmatched reinstatements after cleaning). Term month-60 outcomes (renewal vs expiry) are reported separately from lapse, reinstatement incidence is 7.4% of lapse events.

4. Mortality experience results

4.1 Whole book

Metric	Value
Exposure (policy-years)	2,413,501
Claims (count / amount bn KRWk)	2,321 / 329.62
Expected (count / amount bn KRWk)	3340 / 463.93
A/E amount basis	71.0%
A/E count basis (95% CI)	73.9% (70.9% – 77.0%)

Count A/E exceeds amount A/E by ~3pp — larger policies die less often (see the high-sum-assured cut). Both sexes and both products run in a narrow band: males 74.2% / females 71.1% (count basis); term 76.2% / whole life 72.2% — the product gap is concentrated in the post-renewal durations (Section 4.5). [IN-01]

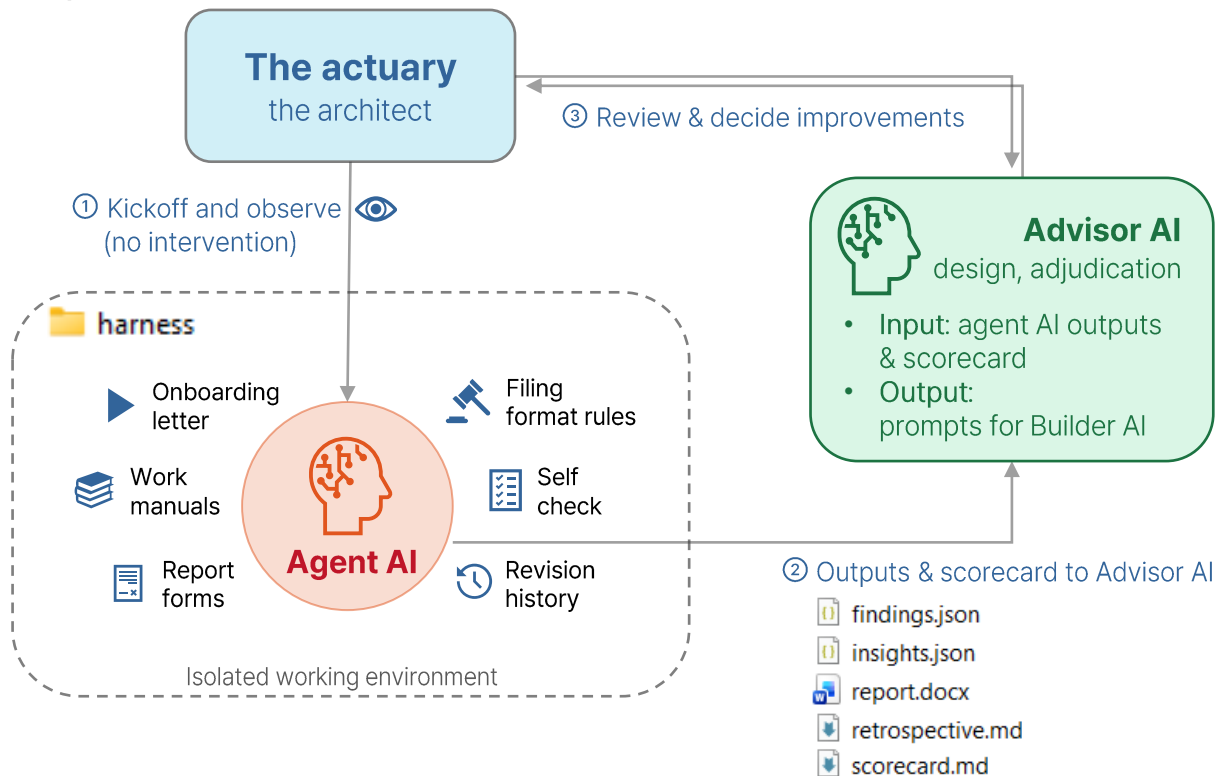
4.2 Duration and trend

Policy year	Claims	Policy-years	A/E amount	A/E count	95% CI (count)
PY1	477	833,677	48.0%	50.3%	0.46–0.55
PY2	494	574,395	69.2%	69.6%	0.64–0.76
PY3	514	412,026	88.8%	93.1%	0.85–1.02
PY4	382	287,731	87.6%	90.8%	0.82–1.00
PY5	266	188,150	85.6%	88.7%	0.78–1.00
PY6	138	86,682	81.5%	80.8%	0.73–1.06
PY7	50	29,840	86.2%	88.6%	0.66–1.17

Selection wears off over approximately two policy years; from year 3 the book runs at 88–93% of the reference table with overlapping intervals — the mature ultimate level on current evidence. [IN-02]

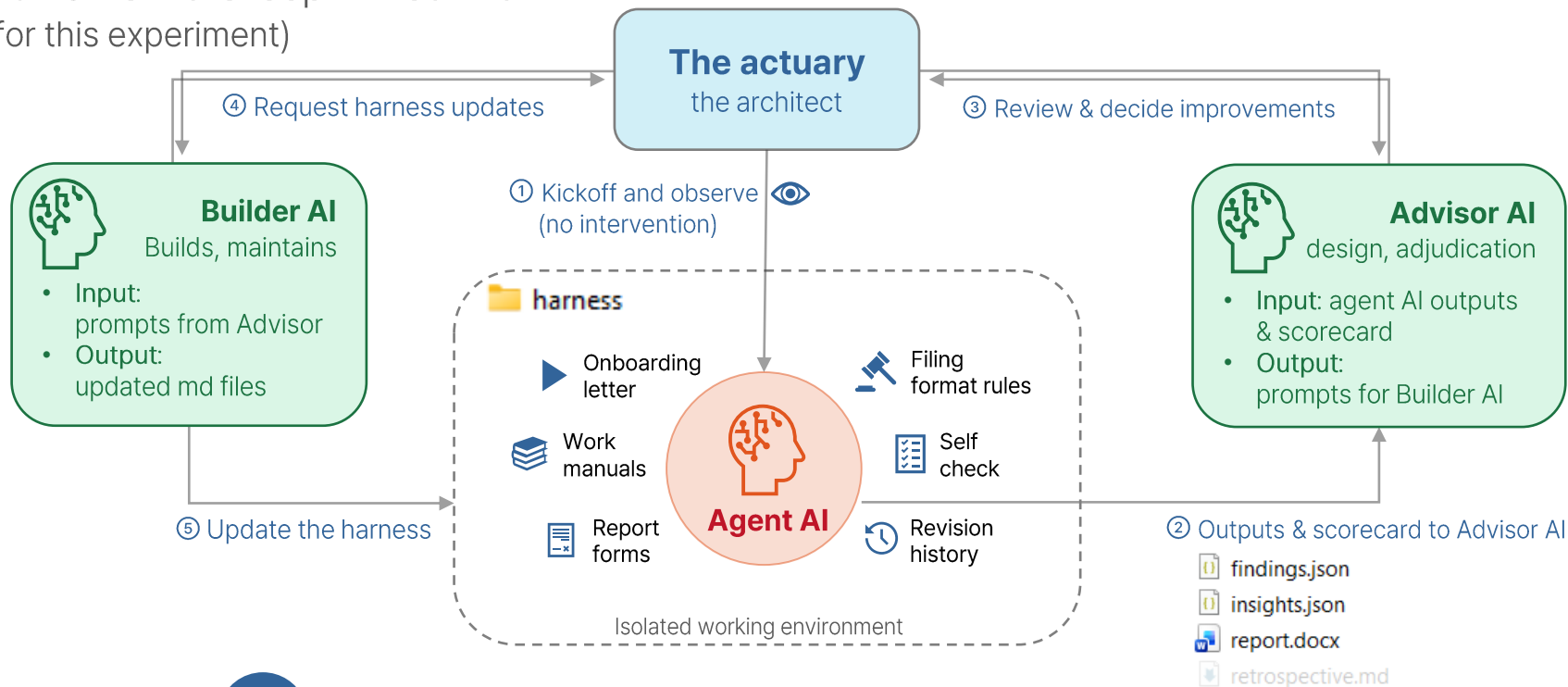
The study cycle

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



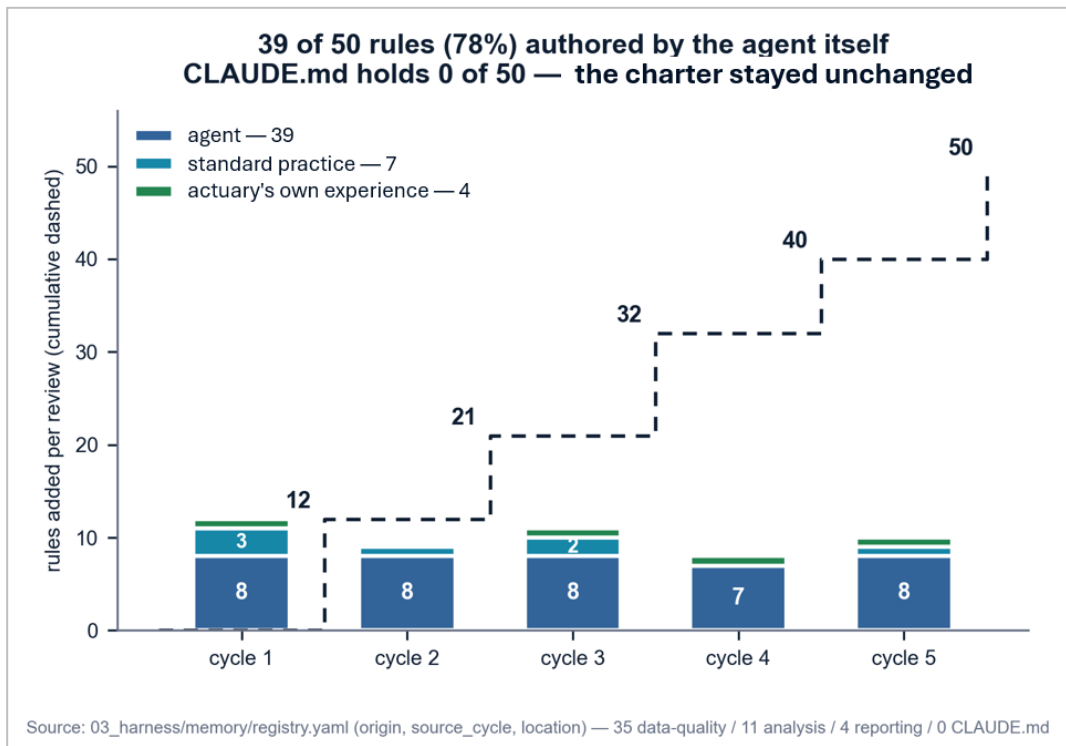
The study cycle

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



 "Every run makes the harness richer"

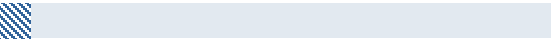
The harness the agent wrote



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.

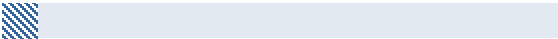


THE AGENT IMPROVED THE ANSWER KEY

"Unrepairable" — repaired.

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

Our answer key said: exclude.

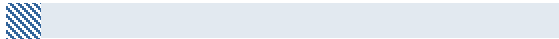


THE AGENT INVENTED A TECHNIQUE

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%.

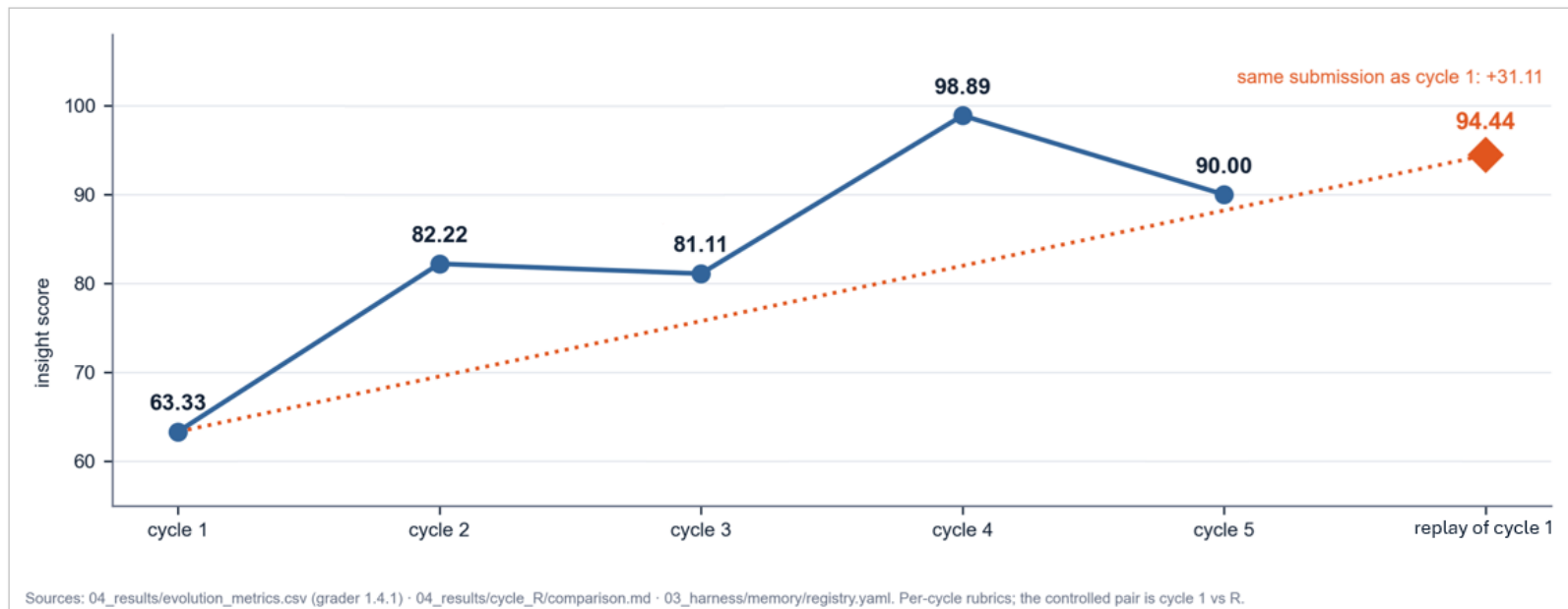
Two defect types it had never seen before. Both found and fixed on first sight — and nothing was flagged that wasn't a problem.

Then it proposed the fixes as new rules.

Scores across cycles

Judgment is where the harness made the difference

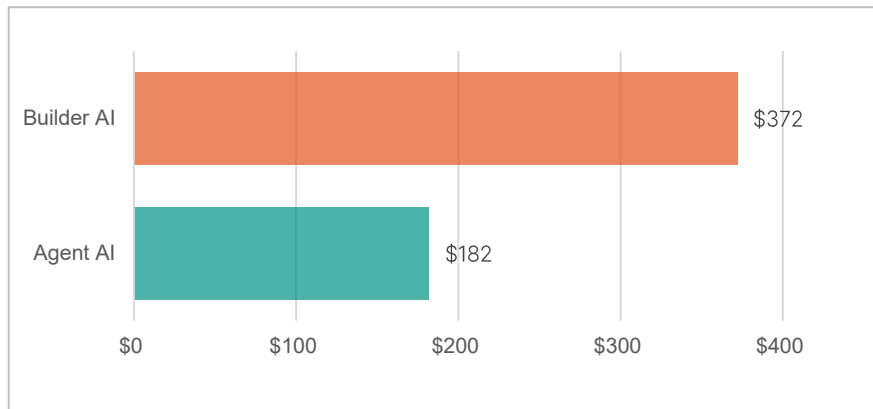
The wiggle is run-to-run noise. The climb is the harness



What did it cost?

The whole experiment, priced at API list rates

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



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

- \$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 $\approx 2\times$ Agent.
The investment sits in the environment — and the environment is reusable.

From Analyst to Harness Architect

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

Building the Test Environment

Synthetic data with known defects, ~31M records



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

Building the Test Environment

Synthetic data with known defects, ~31M records

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	STANDAR	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	STANDAR	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	STANDAR	GA	240000	FALSE	INFORCE	
8	P2019010000007	WHOLE_LIF	1/1/2019	55	F	NEVER	STANDAR	GA	100000	FALSE	LAPSE	10/1/2021
9	P2019010000008	TERM_5YR_	1/1/2019	68	M	CURRENT	STANDAR	GA	70000	FALSE	DEATH	6/26/2019
10	P2019010000009	WHOLE_LIF	1/1/2019	56	F	NEVER	STANDAR	GA	60000	FALSE	LAPSE	1/1/2019
11	P2019010000010	TERM_5YR_	1/1/2019	44	M	FORMER	STANDAR	GA	210000	FALSE	LAPSE	4/1/2019

Policy Events (~1.4M)

	A	B	C
1	policy_id	event_type	event_date
2	P20190100	ISSUE	1/1/2019
3	P20190100	LAPSE	12/1/2019
4	P20190100	ISSUE	1/1/2019
5	P20190100	ISSUE	1/1/2019
6	P20190100	LAPSE	5/1/2019
7	P20190100	ISSUE	1/1/2019
8	P20190100	LAPSE	7/1/2021
9	P20190100	REINSTATEMENT	8/1/2021
10	P20190100	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	P20190100	201901	34	1	240000	10.26
3	P20190100	201901	38	1	30000	0.99
4	P20190100	201901	58	1	490000	115.59
5	P20190100	201901	38	1	50000	2.5
6	P20190100	201901	26	1	190000	6.35
7	P20190100	201901	50	1	240000	28.2
8	P20190100	201901	55	1	100000	8.75
9	P20190100	201901	68	1	70000	44.33
10	P20190100	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	P20190100	6/26/2019	201906	68	1	DISEASE	70000
3	C0000002	P20190700	11/21/2020	202011	59	2	DISEASE	240000
4	C0000003	P20210400	4/16/2021	202104	47	1	DISEASE	510000
5	C0000004	P20190700	3/11/2022	202203	72	3	COVID19	440000
6	C0000005	P20210200	5/17/2022	202205	68	2	DISEASE	170000
7	C0000006	P20200900	7/5/2022	202207	42	2	ACCIDENT	40000
8	C0000007	P20200900	8/3/2022	202208	52	2	COVID19	40000
9	C0000008	P20201000	9/17/2022	202209	62	2	DISEASE	150000
10	C0000009	P20220600	9/25/2022	202209	53	1	DISEASE	280000