Lab 12: Measure it, change it, measure it again

AI Administrator · Day 6 · Session 12 · Monitoring, Evaluation and Continuous Improvement

Participant lab handout

Objective

Instrument your Module 10 triage workflow so that every AI call writes one row to an **AI log** and every failure lands in a **Needs human** queue instead of being lost. Then score the classifier against a **golden set** of twenty e-mails, change the prompt once, score again, and record the decision in a **changelog**. Finish with the judge, the one-sheet dashboard and the kill switch.

Time: 30 minutes in breakout rooms, parts A to D. **Deliverable:** at the end.

What you need

- Your **n8n Cloud** trial from Session 11 (it lasts about 14 days from the day you opened it) with Google Sheets and Gmail connected. No trial? Open one now: *n8n.io, Get started*, sign in with Google, then *Overview, Create credential* for *Google Sheets OAuth2 API* and *Gmail OAuth2 API* (tick every box).
- Your **Module 10 triage workflow** (M10 Triage with approval gate). If it is gone with an old trial, import `Day5_S10_Approval_Gate_Solution.json` from the Session 10 files.
- From the Zoom chat: `Day6_S12_Monitoring_Workbook.xlsx`, `Day6_S12_Golden_Set_Scorer.json`, `Day6_S12_Reply_Judge.json`, `Day6_S12_Send_Test_Emails.json`. The customers, AI Caravan Ltd. and every address are fictional.
- No API key: Gemini 2.5 Flash (and 2.5 Pro for the judge) run on your trial's **Gateway credits**.

Three words for today. A **golden set** is a small collection of real inputs with the answer a competent colleague would give, agreed by two people. A **regression** is a change that makes some cases worse while the total looks fine. **LLM-as-judge** is a second model applying your rubric so you can score hundreds of outputs, with a person checking a sample.

Set-up (5 minutes)

1. Upload `Day6_S12_Monitoring_Workbook.xlsx` to Google Drive, open it, **File, Save as Google Sheets**. Nine tabs: *Dashboard, Golden set, AI log, Needs human, Approval log, Scores, Changelog, Drafts, Judge*. Golden set and Drafts are filled; the rest have headers.
2. **Create workflow**, menu (…), **Import from File**: `Day6_S12_Golden_Set_Scorer.json`. Again for `Day6_S12_Reply_Judge.json` and `Day6_S12_Send_Test_Emails.json`.
3. In every node with a red warning: pick your Google Sheets credential, the **M12 Monitoring** spreadsheet *From list*, and the tab **by name** (the file holds placeholders). In *Send to myself* paste

your own test Gmail address.
4. Open your triage workflow. Confirm it still runs: it is the workflow you instrument in part B.

Part A: two labellers (5 minutes)

Open the **Golden set** tab. Items 1 to 10 are the Session 6 test e-mails you were told to keep; 11 to 20 are new. Cover column F (*agreed*). Label items 11 to 20 in column D; your partner labels them in column E without seeing yours. Then compare both with F. Where you disagreed, the row is probably marked *hard* = yes: those are the cases that decide whether a prompt change helped.

Rule: if two people cannot agree on a case, the model cannot be marked wrong on it. Either agree a label or take the case out.

Part B: log every AI call, fail safely (10 minutes)

1. Start timer and confidence

In your triage workflow add a **Set** node named *Start timer* between the Gmail trigger and *Classify email*: one field `started_at`, type Number, value `{{ Date.now() }}`, **Include Other Input Fields** on.

Open *Classify email*. Replace the last line of the prompt (*Reply with the single word only…*) with:

```
Reply with the label, a vertical bar, and a confidence from 1 (a guess) to 5 (certain),
for example: billing|4
Nothing else: no punctuation, no explanation.
```

In *Clean label* set three fields instead of one: `label` = `{{ ($json.text || '').split('|')[0].trim().toLowerCase() }}`, `confidence` = `{{ (($json.text || '').split('|')[1] || '').trim() }}`, `latency_ms` = `{{ Date.now() - $('Start timer').item.json.started_at }}`, and `raw` = `{{ $json.text }}`.

2. The AI log row

After *Clean label* add **Google Sheets, Append row**, document *M12 Monitoring*, sheet **AI log**, *Map Each Column Manually*, and fill (copy, do not retype):

Column	Value
timestamp	`{{ $now.toFormat('yyyy-MM-dd HH:mm:ss') }}`
date	`{{ $now.toFormat('yyyy-MM-dd') }}`
workflow	`{{ $workflow.name }}`
run_id	`{{ $execution.id }}`
step	`Classify email`
model	`gemini-2.5-flash`

Column	Value
input_200	`{{ ('Subject: ' + $('New email').item.json.subject + ' | ' + $('New email').item.json.text).replace(/[\w.+-]+@[\w-]+\.[\w.]+/g, '[email]').replace(/\+?\d[\d \-]{7,}\d/g, '[phone]').slice(0, 200) }}`
output	`{{ $('Clean label').item.json.label }}`
confidence	`{{ $('Clean label').item.json.confidence }}`
tokens_in	`{{ Math.ceil((580 + ($('New email').item.json.subject + $('New email').item.json.text).length) / 4) }}`
tokens_out	`{{ Math.ceil($('Clean label').item.json.raw.length / 4) }}`
cost_usd	`{{ (tokens_in * 0.30 / 1e6 + tokens_out * 2.50 / 1e6).toFixed(6) }}` with the two expressions above in place of the names
latency_ms	`{{ $('Clean label').item.json.latency_ms }}`
human_decision	leave empty
status	`ok`

Options, Cell Format: RAW. Connect its output to *Draft reply*; in *Draft reply* and *Keep reply* change any `$json.label` to `$('Clean label').item.json.label` (the Sheets node outputs the row it wrote, not your fields).

Run *S12 Send a test e-mail* (i = 0), then **Execute workflow** on the triage. Open the AI log tab: one new row.

What we measured on 28 October 2026: the row estimated 187 tokens in; the Gemini node in the Logs panel reported 179. On the longer draft step the estimate was 30% high (173 against 134). Good enough for a trend line; use the vendor bill for the budget.

3. Retry and the human queue

Open *Classify email*, **Settings**: *Retry On Fail* on, *Max Tries* 3, *Wait Between Tries* 5000 ms, *On Error*: **Continue (using error output)**. The node now has a second, red output.

Connect the red output to a **Google Sheets, Append row** (sheet **Needs human**: timestamp, date, workflow, run_id, step = `{{ $prevNode.name }}`, subject and from from `$('New email').item.json`, error = `{{ $json.error?.message || $json.error || 'see Executions' }}`), then to a **Gmail, Add label** node (*needs-human*, message ID `{{ $('New email').item.json.id }}`).

Test it: open *Gemini (flash)*, change the model name to `models/gemini-9.9-does-not-exist`, send test e-mail i = 1, Execute. Expect three red attempts in about ten seconds, then the Needs human row and the label. **Take the screenshot for the deliverable. Restore the model name.**

> **Trap 1: n8n caps *Wait Between Tries* at 5 seconds.** “Retry after 5, 15, 45 s” is not possible in the node; three tries take about ten seconds in total. Good for a busy endpoint; for an outage the human queue is the real safety net.
>
> **Trap 2: never retry a send.** *Send reply* and *Ask for approval* keep *Retry On Fail* off. A send may already have succeeded before the error arrived; retrying blindly sends twice. Retry the read, then decide again.

Part C: score, change, score again (10 minutes)

4. Score v1

Open **M12 Golden set scorer**. **Execute workflow**: twenty Gemini calls, about 25 seconds. The **Scores** tab gets twenty rows (version, item, expected, got, confidence, correct); the **Changelog** gets one row with the total. Post your score in the main chat.

Our result (28 October 2026): v1 18/20. It called the conference e-mail (item 9) *sales* and the paid-but-locked account (item 19) *billing*. Both are marked hard. Confidence was 5 on every item, including the two wrong ones.

5. One change

In *Prompt version* set `version` to `v2` and `change` to one line describing it. Open *Classify* and replace the rules block with the v2 text from the sticky note (sharper definitions, one example per category, none from the golden set). Change nothing else: not the model, not the temperature.

6. Re-score and decide

Execute again. Changelog: a second row. In Scores, sort by item and compare v1 with v2 line by line: which cases improved, which broke? Then fill `kept` (yes/no) and `by` in the Changelog.

Our result: v2 20/20, items 9 and 19 fixed, nothing broken, kept = yes. **The honest caveat:** v2 was written after reading v1’s two failures, so part of 20/20 is the rules describing this set. The test that matters is next month’s e-mails: add real ones from the Approval log as they come, and re-score.

Part D: judge, dashboard, kill switch (5 minutes, finish at home if needed)

7. The judge

Open **M12 Reply judge** (Gemini 2.5 Pro, the four-check rubric as extractor fields). Execute. The **Judge** tab gets six rows: four yes/no checks, the judge’s total, the human total, *agree*, a one-line reason. Where it disagrees with the human score, read the reason and decide who is right.

Our result: 4 of 6 agree. Draft 3 (jargon): the judge also failed *answers the question* and *safe to send*, stricter than we were. Draft 5: the customer asked for a demo *next week*; the judge called “propose slots next week” an invented promise. A person would not. That is why a person samples the judge.

8. The dashboard

Open the **Dashboard** tab. The four tiles and the 30-day table are formulas over *AI log*, *Needs human* and *Changelog*; today's rows are already counted. If the two charts did not survive the import: select A7:C37, **Insert, Chart**, line chart.

9. The kill switch, the owner, the budget

In cell **B1** write: how to stop the workflow in under a minute (*Overview*, the toggle to *Unpublished*, or open it and *Unpublish*), who may do it including out of hours (names or roles), and the date you tested it. Test it now: unpublish, then publish again. Rows 40 and 41: the owner who reads this tab every Monday; the monthly budget and the 80% alert. Copy B1 into section 7 of your Session 11 policy.

Troubleshooting

Symptom	Fix
Scores shows `correct = 0` on every item	Column F (*agreed*) empty, or spelt differently. Fill all 20 rows in lower case.
Read golden set returns 22 items, two fail	The note rows under the table; check *Only golden rows* is connected.
Classify answers a sentence, not `label|confidence`	The prompt's last two lines were lost when pasting v2. Paste the whole sticky.
Wait Between Tries will not accept 10000	n8n caps it at 5000 ms (Trap 1).
The red output does not exist	Settings, *On Error* must be *Continue (using error output)*.
Add label fails: `labelId not found`	Pick the *needs-human* label From list; create it in Gmail if missing.
`Sheet with ID gid=0 not found`	Pick the document From list again, then the tab by name.
Dashboard chart is empty	Your AI log has today's rows only; one or two points is correct.

Clean up: delete the `[S12 test]` e-mails and the approval request in Gmail; restore the model name.

Deliverable: M12-monitoring

1. The **M12 Monitoring** workbook (export as .xlsx or share the Sheet): Golden set with your labels in D and E, AI log with at least one real row, Changelog with two scored runs and a `kept` decision, the Judge tab, the Dashboard with the kill switch in B1 and an owner in row 40.
2. One screenshot of the failure branch running (three red tries, then *Needs human*).
3. The exported JSON of your instrumented triage workflow (menu …, **Download**).