

WORKED CASE + EDITABLE BUDGET + ACCEPTANCE TESTS

n8n or Make: start with a business task

A working pack to compare a workflow before buying or deploying it.

A maintenance company receives requests through a form. Its team copies the need into the CRM, checks the contact and prepares an intervention. The pilot must produce a reliable, traceable draft, with human approval before writing it.

You will leave with a pilot scope, a monthly budget and a list of evidence to request from the implementer. Use them to decide whether to test, fix or reject an option.

Start with three actions

- 1 Read the worked case**
Check that the outcome resembles what your team needs.
- 2 Replace the yellow inputs**
In the workbook: your volume, time, costs and requirements.
- 3 Request the seven tests**
Keep outputs, logs and billed units for each tool.

Fictional teaching case DEM-104. Amounts illustrate the calculation: they are neither n8n/Make prices nor customer results. No executed comparison of these platforms is presented.

A worked case, from input to output

This is a fictional message. Identity, category and record version come from a test dataset. AI proposes an extraction; the workflow validates its fields before requesting approval.

“Hello, our reception computer will not start. Could you arrange a visit on Tuesday morning at our Rouen Centre site? Please confirm the time slot.”

Reference	DEM-104, received on 3 October 2026, test form
Contact	C-042, exactly one match in the test dataset
CRM record	Version 7; Rouen Centre site
Missing information	Exact Tuesday date, diagnosis, duration, price, availability

Draft field	Expected value and reason
external_reference	DEM-104, retained to find the action after an error
contact_id / version	C-042 / 7, supplied by the test CRM, not guessed by AI
Subject	Reception computer unavailable - Rouen Centre
Requested time	“Tuesday morning”, retained as the customer’s wording
Confirmed date / price	Unset: the message does not establish either
Status	Awaiting approval; no appointment promised to the customer

The manager reviews the request and approves proposal p1 for record version 7. Immediately before writing, the workflow checks this approval and reads the version again. A changed version requires fresh approval. The expected output is one CRM draft with DEM-104 and an approval trail.

Completion means finding the draft, its reference, the approved version and the approving person. A green indicator in the tool is insufficient. This pilot sends no customer email.

What must happen when something fails

The CRM creates the draft, but its response is lost. The workflow does not yet know whether the write succeeded. Immediately retrying a creation can produce a duplicate.

- 1 Look up DEM-104**
Query the state actually stored in the CRM.
- 2 One action found**
Verify its fields and attach the result to the log.
- 3 State remains unknown**
Route the request to review and notify the owner.

Two simultaneous retries may both find no existing record. A lookup before creation therefore does not guarantee uniqueness. Require an atomic mechanism at the destination (an idempotency key or unique constraint), or coordination with demonstrated protection against concurrent retries.

In Make, Rollback applies to transaction-compatible modules and depends on the auto-commit setting. It is not a universal undo button for every application. Deleting a record does not necessarily undo a message already sent. [3]

Give the backup operator the request location, how to read its state, who may retry and the evidence needed to close an incident. Ask that person to recover a case without help from the workflow author.

Which option should you test first?

This is a conditional testing order, not a measured ranking. An unproven mandatory requirement suspends selection even when the price is attractive.

Your constraint	First assessment	Condition for proceeding
The team already runs Make	Make	Verify the exact CRM action, permissions and seven tests. Familiarity alone does not validate the workflow.
The team already runs n8n	n8n, in its current mode	Use the same tests. Verify operational ownership and backup cover.
No existing tool or dedicated operator	Make and n8n Cloud	Compare two hosted services on the same business result and full cost. [1]
Orchestration must run in your environment	Self-hosted n8n	Name an operator, test backup/restore and review every transfer to an external model. [1]
A mandatory action or recovery is missing	Pause this candidate	Cost and test an adaptation. Do not average away this missing requirement.

Worked DEM-104 decision: the fictional team already knows Make and accepts a hosted service. It starts with a Make pilot. Deployment remains conditional on test evidence and observed cost. If the CRM action or recovery fails, fix and retest, or assess n8n Cloud against the same criteria. This reasoning does not prove that Make is superior.

A cheaper subscription can require more operational work. In this example, saving €60 but adding 2 hours at €60/hour increases monthly cost by €60. These are fictional values.

The cost that changes the decision

Same monthly scope: 1,000 requests, all reviewed. Review time includes approval and ordinary corrections; do not count those minutes again as technical maintenance.

Fictional item	Calculation	€/month
Human review	$1,000 \times 30 \text{ s} \div 3,600 \times \text{€}36/\text{h}$	300
Service + model + hosting	$60 + 20 + 0$	80
Technical maintenance	$2 \text{ h} \times \text{€}60/\text{h}$	120
Recurring cost	$300 + 80 + 120$	500
Setup spread over 12 months	$\text{€}1,200 \div 12$	100
Full cost on this basis	$500 + 100$	600

At 90 seconds per review, human review costs €900. Full cost becomes €1,200/month. If the previous manual process took 2 minutes at €36/hour, the value of that working time was also €1,200. The theoretical benefit disappears within this scope.

Available time is not guaranteed cash savings. It must be reassigned or reduce actual spending. Make credits, n8n executions and model calls are not interchangeable units. Measure them separately. [1, 2]

First change the yellow workbook assumptions. Recurring cost and cost including setup recalculate. Missing inputs return "n.a.". Save a separate copy with the other option's own observations.

Acceptance tests for the implementer

Test	Expected outcome
T01 - valid request	One correct draft after approval; retrievable reference and evidence.
T02 - ambiguous identity	No write; request routed to review.
T03 - API down before write	Visible failure, no creation, controlled recovery.
T04 - repeated event	One action per reference, including concurrent attempts.
T05 - changed record	Approval becomes stale; fresh approval required.
T06 - response lost after creation	Find the existing action; no blind second creation.
T07 - missing or stale approval	No write until valid approval covers the current proposal.

For each tool record version/plan, date, final state, human time, service units, model cost and an evidence link. Use fictional data and reset state between independent cases. T04 and T06 must retain state between attempts.

Proposed decision: proceed when all mandatory requirements are demonstrated, cost is accepted and the backup operator is independent; fix and retest if a point remains open; reject if the constraint cannot be met. "Not tested" never means passed.

Sources and inspiration

Original Initial IA pack. Make templates informed the requirement for a concrete example with instructions; no import-ready scenario or US guide is reproduced. The technical sources document features, not performance. Consulted on 3 October 2026.

[1] n8n: plans, Cloud and self-hosting

[2] Make: credits and consumption modes

[3] Make: Rollback limitations

[4] Make: template library