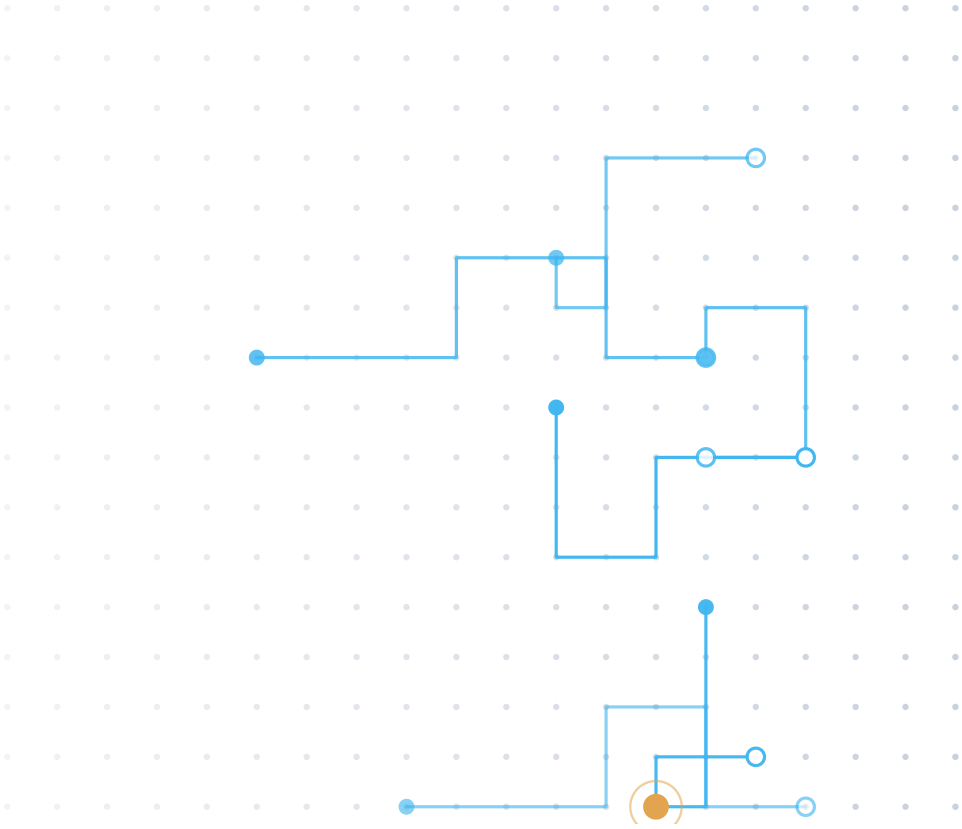


PRACTICAL GUIDE

AI agents in production without silent failures

How to know every morning, with evidence, what your agents did and what they left undone.

AUTHOR

Mario Chasco · Flowara AI

FOR

Business owners who already run AI agents and the technical people they hire to build them

READING

9 min

EDITION

v2.0 · November 2026

BEFORE YOU START

What you will get from this guide

01 Spot the failures that never raise an error, and the clue each one leaves.

02 Put a log, a count and a canary around any agent.

03 Decide with tests which type of task can go out without review.

04 Ask for a morning report that also says what was not done.

CONTENTS

01 Silence does not mean everything is fine
02 The four typical silent failures
03 Where they start: four gaps in the build
04 The rule of three: log, count, canary
05 Autonomy is earned task by task
06 The morning report and who reads it
07 Example: a neighbourhood hair salon with a booking agent
08 Production checklist

For: Business owners who already run AI agents and the technical people they hire to build them · 9 min read

01

Silence does not mean everything is fine

A smoke alarm with a dead battery is just as quiet as a working one in a house with no fire. From the sofa you cannot tell them apart. AI agents are the same: the one that did nothing and the one that did everything right make exactly the same amount of noise, none.
An error that pops up on screen is annoying, but it gets fixed that morning. The dangerous one is the error that never speaks up. The report says "nothing pending" because it could not read the inbox, the scheduled job has not run for three weeks, or a message went out half-written. You find out when a customer complains, and by then the damage is done.

This guide is for the business owner and for the technical person who builds the system. Both need the same thing: to check every morning, in two minutes, that the agent did its job and what it left undone. This is what I set up before any agent works on its own, and here it is piece by piece.

> ✦ IDEA
>
> An agent without an error log is like an employee who stopped turning up without anyone noticing.

02

The four typical silent failures

Almost every failure that goes unnoticed fits one of these four. None of them shows an error. Each one leaves a clue, if you know where to look.

FAILURE	WHAT IT LOOKS LIKE	WHAT GIVES IT AWAY
The report says zero and it is false	"No pending messages"	Items-read counter at zero
The job that never ran	Nothing: no error, no report	Last run older than scheduled
The half-sent message	The customer gets half a reply	A "sending" status that never becomes "sent"
The stale context	Old prices or opening hours	A review date on every fact

The first is the most treacherous, because "I found nothing" is not the same as "I could not search". If email access has expired, the agent reads zero messages and calmly concludes there is no work. That is why every lookup has to return one of three answers: **found, not found or failed**. And "failed" never counts as success.

The last one creeps up slowly. Nobody tells the agent that the summer opening hours no longer apply, so it keeps offering them with total confidence until someone checks.

IN ONE LINE

Zero read is not zero work: make the system tell the two apart.

03

Where they start: four gaps in the build

The failures in that table rarely come from the model itself. They come from gaps in the code around it, and nearly always from these four:

01 **A reply nobody can read and nobody receives.** The model returns something the code cannot parse, the step fails and the message ends up nowhere. Every unreadable reply has to land in a person's inbox.

02 **A lookup with an empty key.** If the customer's phone number arrives blank, the search may return nothing, or return someone else's record. Before searching, the code checks that the key exists and has the right format.

03 **A tool that cannot say "not found".** If the calendar lookup only knows how to return free slots, a fully booked day looks like an error. Worse, the agent may fill the gap with something it made up. Every tool needs an explicit way to answer "there is none".

04 **Two automatic replies talking to each other.** The agent answers an out-of-office message and the other side answers back, all night long. Automatic replies have to be recognised, and every conversation needs a cap on the number of messages.

> ⚠ **WARNING**
>
> None of these four shows up in a demo with tidy examples. They appear with real messages, so they go into the tests from day one.

IN ONE LINE

The model writes the reply; the code around it is what has to fail loudly.

04 The rule of three: log, count, canary

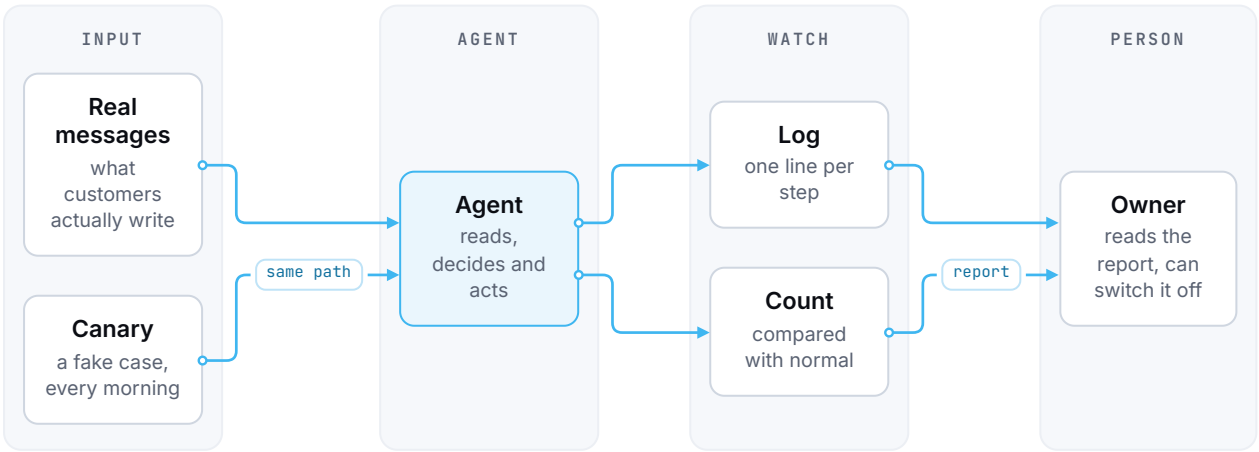
Every agent that works unsupervised needs three pieces. None of them is expensive or complicated. Together they catch almost everything that table lets slip.

01 **Log.** One line per step: what came in, what was decided and what was done, with personal data masked. And a named person who reads it, because a log nobody looks at is not watching anything.

02 **Count.** How many items it read, how many it processed, how many it handed to a person and how many failed, compared with what is normal. A Monday with zero messages read is an alarm, not a quiet day.

03 **Canary.** A fake, known case that comes in every day through the same path as the real ones and has an expected result. If the canary does not come out as it should, something is broken even if nobody has complained.

FIGURE 1 **Three pieces around one agent**



EXAMPLE

Every morning a test message from "Canary Customer" asks for an appointment on a day the business is closed. The right answer is to offer another day. If it does not show up in the report, the system raises an alert.

One more check completes the set: if the most recent run is older than the scheduled time, an alert goes off. That is the one that catches a job that quietly stopped running.

IN ONE LINE

If one of the three pieces is missing, a failure can stay silent.

05

Autonomy is earned task by task

An agent is not "trusted" as a whole. You trust it **for one type of task**: answering frequent questions, yes; changing an appointment, not yet. That decision lives in configuration, not in the model's instructions, so nobody changes it by accident while editing a prompt.
Everything starts as a draft. The AI writes; a person approves. Every correction is saved as a written rule, and the case joins the tests. A task type moves to automatic sending only when every test passes and the owner asks for it.

The tests are a set of **reference cases** (often called a *golden set*): anonymised real conversations, including hard cases and attempts to trick the agent. Each case is run several times, five for example, because a model can get a message right once and wrong the next time. One failure is enough to keep that task type in drafts.

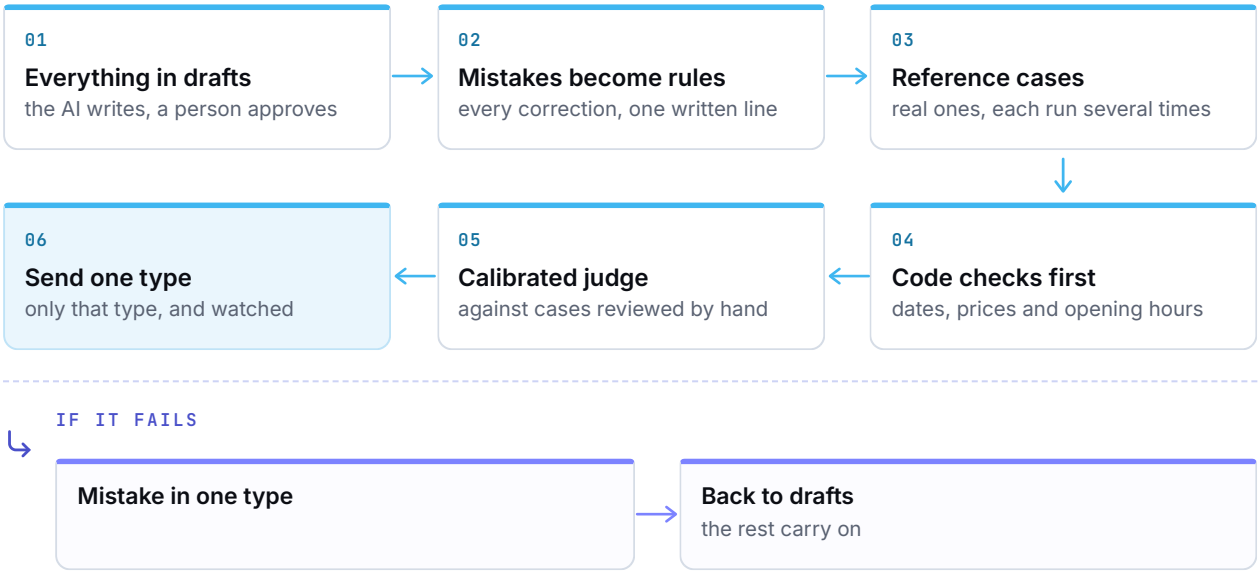
> *What a rule can check, the code checks; the model only judges what needs reading.*

FLOWARA AI DESIGN PRINCIPLE

In practice, code first checks that the date exists, that the price matches the price list and that the time falls within opening hours. What needs actual reading, such as tone or whether the reply really answers the

question, is judged by another model. That judge is calibrated too: before you rely on it, compare its verdicts with cases a person has already reviewed by hand.

FIGURE 2 **From drafts to automatic sending**



IN ONE LINE

Autonomy is proven with tests per task type, never on gut feeling.

06 The morning report and who reads it

The daily report is the contract between the agent and its owner. To be believable, it needs three things:

01 **Counters:** read, processed, sent, handed to a person and failed.

02 **Differences:** every number next to what is normal. "12 failed" tells you nothing; "12 failed, normally 0" does.

03 **What I did not do:** what it could not read, what it left as a draft and why. If reading was incomplete, the report says "coverage incomplete", never "nothing pending".

The difference is clearest when the same day is reported in two ways:

TABLE 1 **The same Tuesday, two reports**

	No counters	With counters
INBOX UNREACHABLE	"Nothing pending"	"Read 0, normally 30-45"
JOB DID NOT RUN	Nothing arrives	Alert: no run at 07:00
CANARY	Does not exist	"Canary: missing"
PARTIAL READING	Not mentioned	"Coverage incomplete"
DRAFTS	Not mentioned	"4 appointment changes need approval"

A report that works fits in three lines:

```text
Mon 07:00 · read 41 (normal 35-50) · sent 28 · to a person 9 · failed 0
Canary: OK · Coverage: complete
Not done: 4 appointment changes left as drafts (need approval)
```

On top of that, every agent has **a named person** who reads that report, decides what to do about failures and can switch it off. "The team" or "the vendor" will not do: it has to be a name. You can delegate the work, but not the understanding of what is going on.

IN ONE LINE

A report that does not say what was left undone does not deserve your trust.

07 Example: a neighbourhood hair salon with a booking agent

A neighbourhood hair salon uses an agent that answers the business's messages: opening hours, prices, new appointments and changes. This is a made-up example to illustrate the point.

Week 1. Everything is a draft. The owner approves every reply from her phone and corrects the same mistake three times: the agent was offering Monday slots, and the salon is closed on Mondays. That correction becomes a written rule and a test case.

Week 3. Frequent questions pass every test, with each case run several times, and the owner asks for them to go out on their own. The split ends up like this:

01 Frequent questions: sent automatically.

02 New appointments: draft she approves.

03 Appointment changes and complaints: always to her.

One Tuesday. The report says "read 0 (normal 30-45)" and the canary is missing. Access to the messaging channel has expired. Without a counter, the report would have said "nothing pending" and the salon would have lost a day of bookings without knowing. Access is renewed, its expiry date is written down and a reminder is set for a week before it expires again.

Notice that nobody had to rebuild the agent. A new rule was enough, plus a date written down with its reminder.

IN ONE LINE

Every mistake ends up as a written rule: that is how an agent improves without being rebuilt.

08

Production checklist

Before you leave an agent working on its own, check these fourteen points. If any is missing, that agent stays in drafts:

Before an agent works alone

14 ITEMS

- [ ] A named person owns the agent
  and can switch it off
- [ ] Every step leaves a log line
  with personal data masked
- [ ] Someone reads the error log every day

- [ ] Counters are compared with what is normal
- [ ] A daily canary travels the real path
- [ ] A scheduled run that does not happen raises an alert
- [ ] "Not found" and "failed" are different results
  neither counts as success
- [ ] Every unreadable reply reaches a person
- [ ] Each conversation has a message cap
  it stops loops between automatic replies
- [ ] No message stays in "sending" without review
- [ ] Every fact and every access has an expiry date
- [ ] Automatic sending is set per task type
  and lives in configuration
- [ ] Every automatic type passed all its tests
  each case, several times
- [ ] The report includes what was not done and the coverage

NEXT STEP

Free diagnostic (20 min)

If you want to see which part of your business can be automated with the risk under control, book a free 20-minute diagnostic.

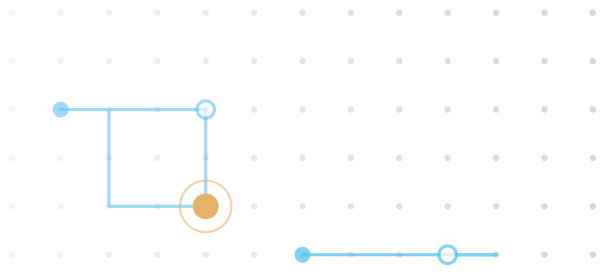
Book a diagnostic ›

flowaraai.com/en/contact

RELATED RESOURCES

Public booking agent template ↗

Real automation cases ↗



Mario Chasco · Flowara AI

AI systems in production for small businesses.

flowaraai.com
github.com/mariochasco78557
youtube.com/@mariochasco_AI
linkedin.com/in/mario-adrian-chasco-frauca

Share this guide as is, crediting flowaraai.com.