

2026 FIELD NOTE

Everyone shipped a pilot. *Almost nobody shipped.*

Why agents stall between the demo
and the risk committee, and what
closes the gap.

The model was
never the
bottleneck.

52%

are piloting agents

Financial services firms, Cambridge CCAF 2026 survey

21%

have any in production

The other thirty-one points are stuck in review

12%

of agentic use cases shipped

Across all industries in the past year

Four questions that kill the *demo.*

01 What exactly did it do?
Not a transcript. A record.

02 What stopped it doing worse?
Name the control, not the prompt.

03 Who approved this version?
And what changed since.

04 Prove it, six months from now.
To an auditor who was not there.

The deadline moved. The *evidence bar* did not.

EU AI ACT

Annex III high risk, which includes creditworthiness, deferred to **2 Dec 2027**. Article 50 transparency and Article 4 AI literacy stayed put.

DORA

Articles 28 to 30. **19 critical ICT providers** designated, under direct EU oversight. Your register entry and exit plan are due regardless.

RBI

FREE-AI, seven sutras, 26 recommendations. The June 2026 model risk draft adds human override and **kill switch** expectations.

SR 11-7

Model risk management. It never stopped applying because the model started talking.

Guardrails are *config*, not prompts.

```
# policies/collections.yaml
pre:
  - block_outside_window:
      hours: "08:00-19:00"
      tz: Asia/Kolkata
post:
  - block_contact_frequency:
      max_contacts: 3
      window_hours: 168
  - require_disclaimer: ai_agent
never: [threats, third_party_disclosure]
```

A prompt is a request. A rule is a
control: versioned, testable, diffable
in a pull request, and enforced whether
or not the model cooperates.

Blast radius is the *tool list*, nothing else.

```
# tools.py
@zolva.tool
def create_payment_link(loan_id: str) -> PaymentLink:
    """Amount comes from the loan system,
    never from the conversation."""

# there is no approve_application tool.
# that is the whole point.
```

Typed Python functions with Pydantic models. An agent cannot do what it has no tool for, so the review is a list of functions rather than an argument about intent.

Behaviour is a *test suite*, and CI is the gate.

```
# evals/safety.yaml
cohort: hardship_handoff
grader: handoff
min_pass_rate: 0.98

# .github/workflows/release.yml
$ zolva eval --gate
$ zolva synthetics --gate
$ zolva compliance --gate

# red build, no release. no meeting needed.
```

Cohorts of adversarial cases, graded by judge or by handoff. Regression in agent behaviour **fails the build** the same way a broken unit test does.

An audit trail you can hand *to a regulator.*

- 01 Every input, output, and tool call
with the arguments, not a summary
- 02 Every rule that fired, and what it blocked
including the ones that passed
- 03 The config hash in force at that moment
so the version is not a guess
- 04 Hash chained, so a gap is visible
SQLite to start, Postgres in production

Redaction protects the customer. The
chain protects you. Both, or the record
is not worth producing.

Seven jobs that cannot ship without *evidence*.

<div>01</div> <div>Customer support</div> <div>Real balances, no advice, no leakage</div>	<div>02</div> <div>Loan collections</div> <div>Contact windows and frequency caps</div>
<div>03</div> <div>Fraud and disputes</div> <div>File the case, promise nothing</div>	<div>04</div> <div>KYC and onboarding</div> <div>Status and reupload, never approval</div>
<div>05</div> <div>Insurance claims</div> <div>No coverage calls, no estimates</div>	<div>06</div> <div>Patient support</div> <div>Claims and denial codes, not clinical</div>
<div>07</div> <div>Telecom operations</div> <div>Only offer codes that actually exist on the account</div>	

THE SHORT VERSION

Build the agent
in an afternoon.
Spend the rest
proving *what it
cannot do.*

```
$ pip install zolva
```

```
$ zolva validate && zolva eval --gate
```

DOCS AND USE CASES › [ZOLVA.AI](https://zolva.ai)