How To Survive In The AI Jungle

Rethinking Test Strategies For An AI Era

Nicole van Gijn · Tech Lead Testing AI · Creator of Evaluaite

If AI is correct 80% of the time...

Why does that still scare us?

Same Prompt, Different Outcomes

Software never used to behave like this.

Same Prompt
Consistent input leads to varied results

Same Model
Utilizing the Same AI Framework Repeatedly

Different Behaviour
AI actions can shift unexpectedly based on input

Different Answers

Outcomes can vary significantly each time

Air Canada - Chatbot

When behaviour fails in production

Reality Check

When behaviour fails in production

Lesson learned

AI behaviour was not adequately evaluated for hallucinations and unreliable responses.

The AI Danger Zone

Probabilit
y
Outputs can vary every run

Danger Zone
Weakly bound guardrails

Authority
People trust confident answers

Still lost though...

Let’s find our way back with these four shifts

Shift #1

From:Deterministic system To: Probabilistic behavior

Traditional systems:

→ Input

Logic →

Output

Both involve complex decision-making processes

AI systems:

→Input

Context →

Probability

→

Behaviour

Shift #2

FROM:
Pass or Fail

To:
Is it behaving within acceptable boundaries?

Shift #3

FROM:
Single Output Validation

To:
Testing AI behaviour under variation

Shift #4

FROM:
From Test Cases

To:
Quality Intelligence through Metrics

Okay Nicole...

How do you validate behaviour?

A Structured Approach to AI Quality

#1 Core Concept

AI risks become visible through behaviour

#2 Core Concept

AI quality has two layers

Model

Behavior generated by the model

- probabilistic
- context driven
- semantic evaluation

System

Behavior around the model

- deterministic
- engineering driven
- technical validation

#3 Core Concept

Before measuring quality, define quality

Quality dimensions define what good behaviour means.

#4 Core Concept

Metrics make behaviour visible.
Thresholds make it controllable.

Walkthrough

Air Canada chatbot

Observed Behaviour

Air Canada's chatbot promised refunds customers were not eligible for.

Primary Risk

Context / Retrieval Drift

The selection or interpretation of context changes, causing the system to use inconsistent or incorrect knowledge for similar inputs.

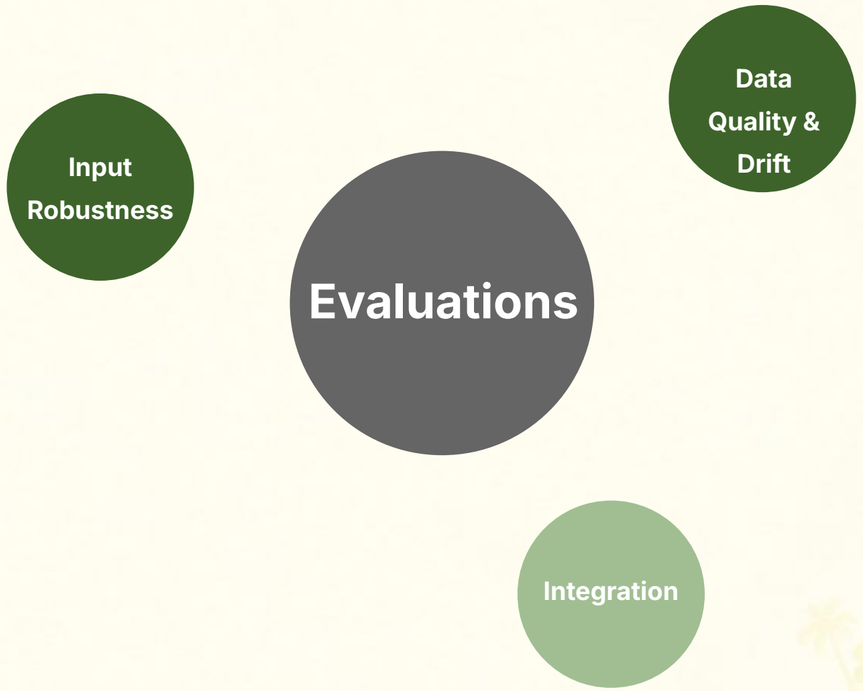
Impact: High

Pillars

Quality lenses

One risk translates into multiple quality areas

- Data Quality & Drift (P)
- Input Robustness
- Integration & SystemQuality



Quality Dimensions

The quality characteristic used to evaluate the observed behaviour.

Quality Dimension	Observed Behaviour
Stability	Small prompt variations produce different refund decisions
Consistency	Similar customer questions receive conflicting policy guidance
Boundary Control	The chatbot responds outside approved refund policy boundaries

Metrics & Thresholds

Not every metric behaves the same.

- High-risk threshold: **0.85 / 0.95**

Quality Metrics (Higher = better)

Metric	What do we measure?	Threshold
Groundedness	Is the response supported by retrieved policy context?	≥ 0.95
Contextual relevancy	Does the chatbot response align with the retrieved policy context?	≥ 0.95

Risk Metrics (Lower = better)

Metric	What do we measure?	Threshold
Hallucination Rate	Percentage of unsupported refund claims	≤ 5%
Behaviour Variance	How strongly do refund decisions vary across similar prompts?	≤ 5%

Golden Datasets

Why it matters

- creates reference behaviour
- enables regression testing
- makes evaluations repeatable
- reduces subjective

Golden datasets create a shared baseline for AI evaluation.

```
golden_dataset = [
    {
        "input": "Can I get a refund after my
flight?","expected_behavior": "deny_refund"
    },
    {
        "input": "I lost my receipt",
        "expected_behavior": "request_documents"
    },
    {
        "input": "My shoes don't fit",
        "expected_behavior": "approve_return_policy"
    }
]
```

From prompts to measurable AI quality

How teams operationalize AI quality

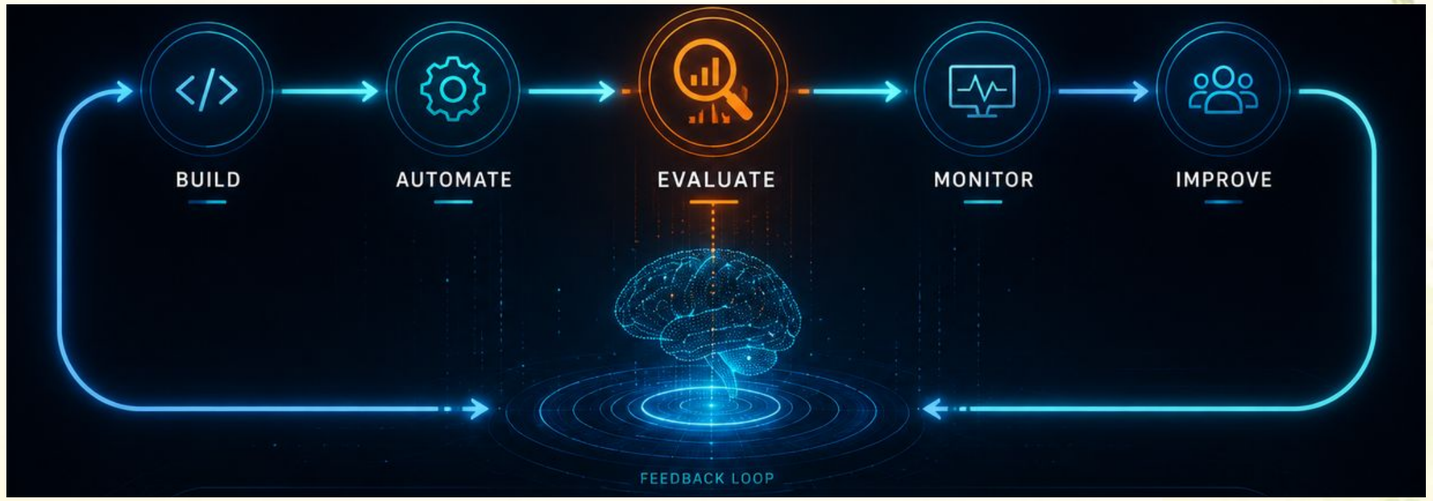
- **Contextual Relevancy**
  - Measures whether the generated response aligns with retrieved policy context.
- **Threshold**
  - Defines where acceptable behaviour ends.
- **Evaluation Result**
  - Makes behavioural drift visible over time.

```
metric = ContextualRelevancyMetric(
    threshold=0.95,
    include_reason=True
)

test_case = LLMTestCase(
    input="Can I get a refund?",
    actual_output=response,

)retrieval_context=policy_context

metric.measure(test_case)
```

From prompts to measurable AI quality

How teams operationalize AI quality with Evaluation Driven Development

- **Evaluations**
  - instead of asserts
- **Thresholds**
  - instead of absolute truths
- **Observability**
  - instead of only logging
- **Continuous validation**
  - instead of one-time testing



How to Survive in the AI jungle

Survival starts when we stop chasing correctness and start navigating behaviour.

Continue navigating the AI jungle

Explore practical tools for building measurable AI quality

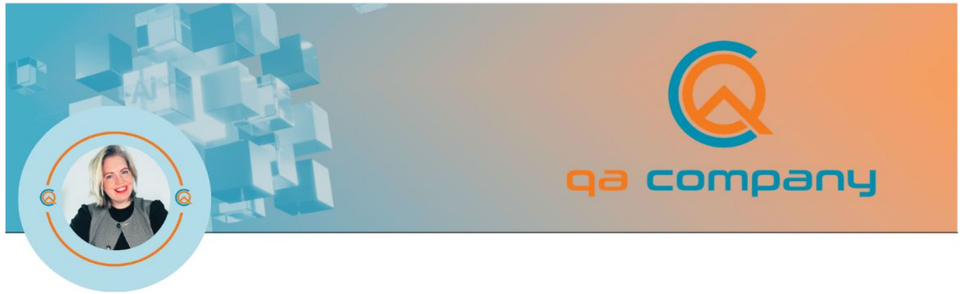
Download evaluaite

Start mapping AI risks into measurable quality

Download me

Follow the journey

AI Testing • Quality Engineering • Evaluation-Driven Development



Nicole van Gijn- Tingelaar

💁‍♀️ Tech Lead Testing AI | Creator of Evaluaite | Building the future of AI Quality Engineering