Cutting Through the Noise

Engineering quality in an AI-based product

James Pearce - Senior AI Quality Engineer

Starting Martial Arts


JOSEPH THOMAS GANNON
WORLD2014
Tatami
IBJJF

The warm up mat… Focus on three things:

Be First

Make the first move. It gives you the initiative.

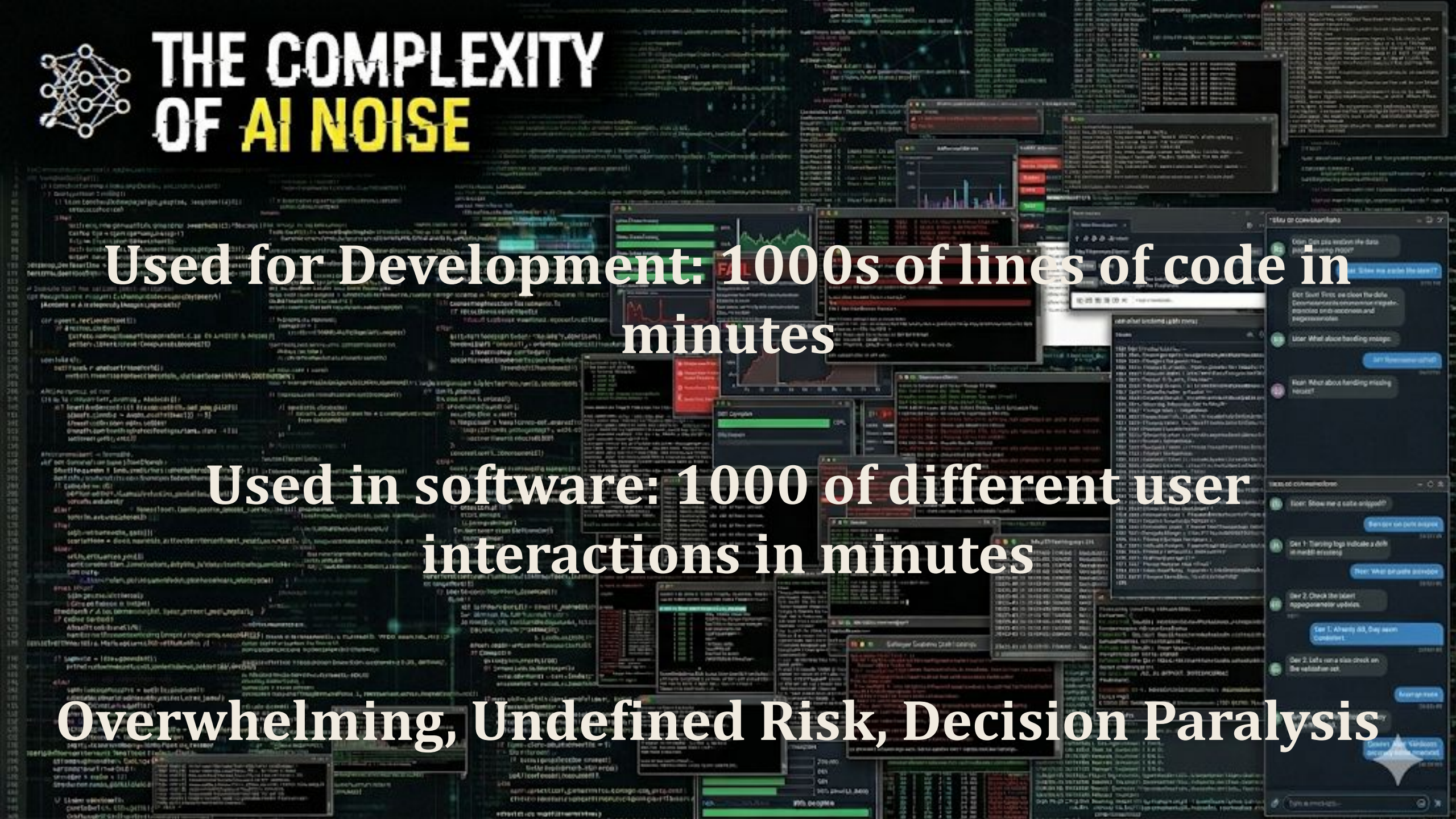
Take Risks

A bad decision beats no decision at all.

Win or Learn

Win and it's great; lose and you'll learn more.

Learnt - I’m not good at having my photo taken


THE COMPLEXITY
OF AI NOISE
Used for Development: 1000s of lines of code in minutes
Used in software: 1000 of different user interactions in minutes
Overwhelming, Undefined Risk, Decision Paralysis

Three concepts for testing AI systems

Verifier's Law

Measured Risk

You Win or You Learn

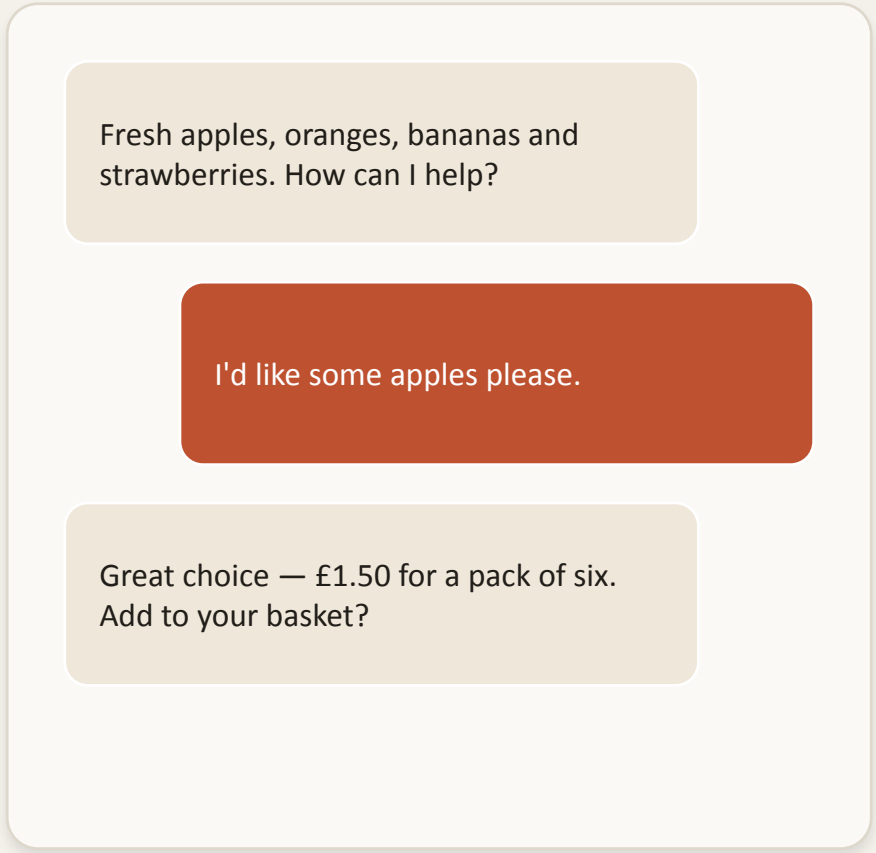
By the end of this presentation . . .

1. **Is it good enough to release?**
2. **How to agree what to fix first.**
3. **How to release changes safely, and learn.**

Meet the fruit selling chatbot.

An AI powered chatbot that sells four fruits: apples, oranges, bananas and strawberries.

- **Sell fruits -** through a chat interface
- **Handle questions -** to ease any concerns
- **Hands off -** to the website basket



Three new requested features

F1

Live prices

When a customer asks the price of any fruit, the bot gives the current price.

F2

Apple upsell

When a customer buys other fruit, the bot suggests adding apples. Everyone likes apples.

F3

Allergy guardrail

Never give allergy or food-safety advice. Any such question gets a fixed redirect.

CONCEPT 1 - Verifier's Law

The speed of development is proportional to how verifiable the task is.

If you can't prove it works, you'll be slow. So design things you can prove.

What is determinism

In order to measure how verifiable something is we work out at how deterministic it is

DETERMINISTIC

Q What's 6 × 60p?

A £3.60

One right answer. A machine checks 10,000 of these a second.

NON DETERMINISTIC

Q Write a friendly greeting.

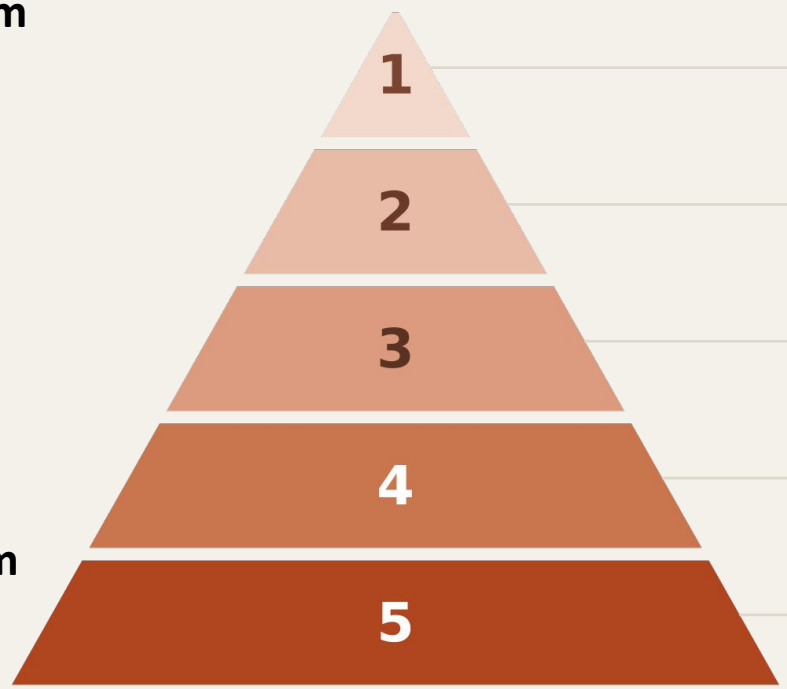
A *…a thousand good answers*

Open to interpretation. based on judgement and two markers can disagree.

The determinism pyramid

A tool we to use work out how deterministic an AI system is.
Each of the layers asks a question about how we use AI in the system

High determinism
Lower risk
Less testing



Low determinism
Higher risk
More Testing

Just use code
"6 × 60p" → == £3.60

Structured output
Extract {fruit, qty} → fields match

Tool call
Get_price() → matches the till (the key)

Constrained output
"≤ 10 words, yes/no" → format + contains

Measure success
"Write a friendly greeting" → humans rate a sample

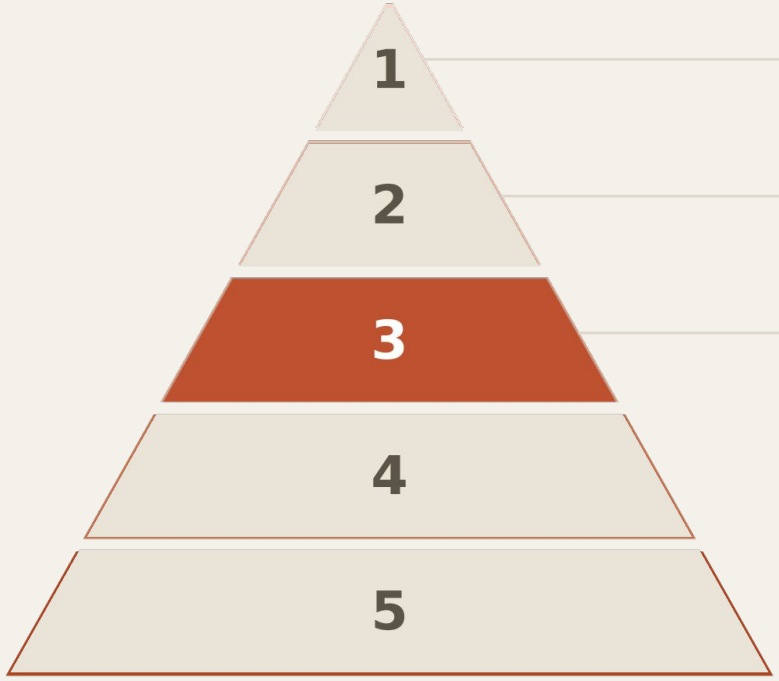
Feature 1 - Live Prices

INSTINCT — PROMPT-BASED

“Bananas are 60p, oranges are 80p.”

Quick to ship but it leaves the maths to the AI. Verification would require 10000s of every fruit, quantity and phrasing?

OUTCOME - We can evaluate when the tool is called, ensure what comes back is valid and verify the AI uses it correctly - 2 Days of testing



Just use code

NO it needs to be conversational

Structured output

NO there is no source of truth in the conversation

Tool call

YES we can call out to an external source of information for the prices instead of using AI to generate them

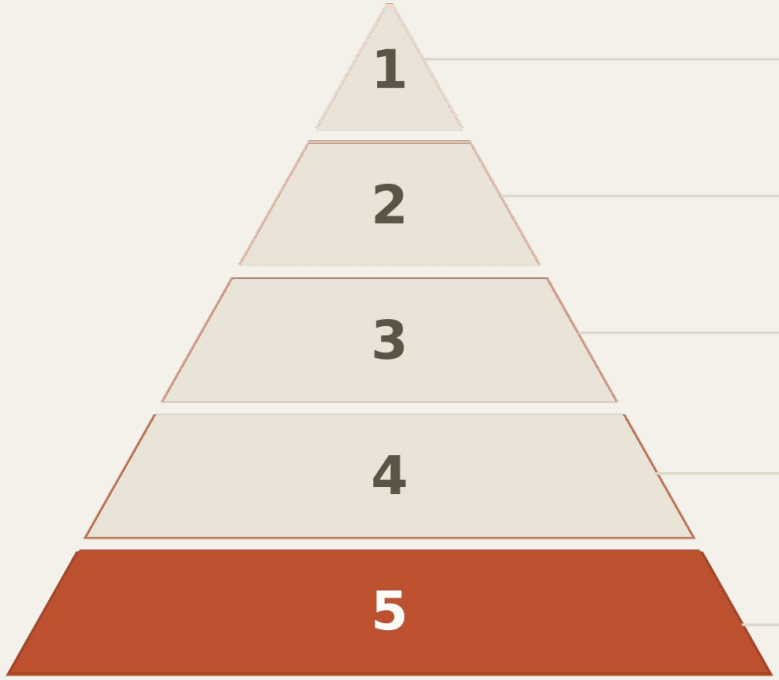
Feature 2 - The apple upsell

INSTINCT — PROMPT-BASED

“Suggest apples when appropriate.”

Seems to work in a few tries but the AI has no intuition for “appropriate”, and there's no assertion for “suggested naturally”.

OUTCOME - We need 500+ Human labelled test cases to evaluate the change - 1 Week of testing



Just use code

NO it needs to be conversational

Structured output

NO there is no source of truth in the conversation

Tool call

NO theres is no source of truth for “where appropriate”

Constrained output

MAYBE but it would abstract the decision to another call with the same problem

Measure success

YES We need to try to measure how effective it is

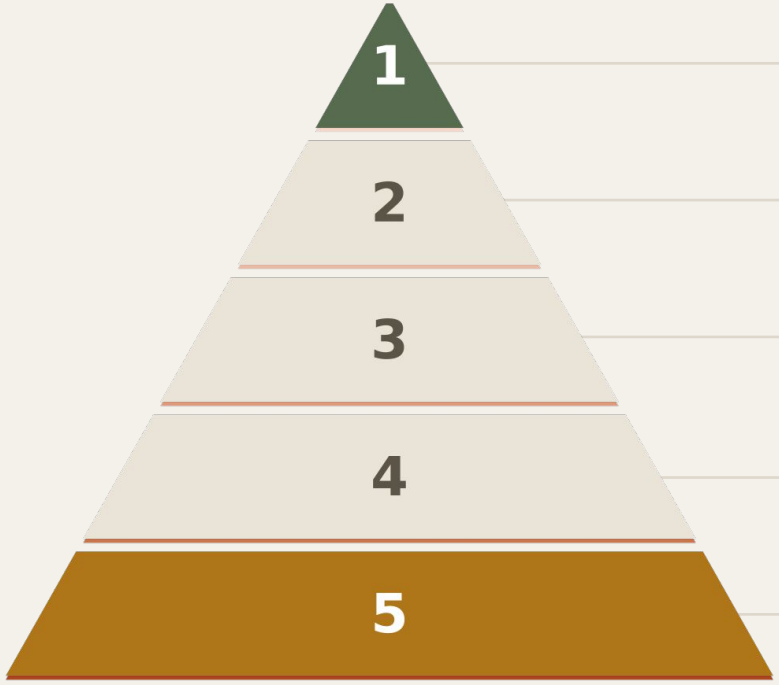
Feature 3 - Allergy guardrail

INSTINCT — PROMPT-BASED

Instinct: “Never give allergy advice”

Quick to ship but there are 1000s of ways to ask about allergies and 1000s of ways the AI could answer that could be wrong or dangerous.

OUTCOME - We can evaluate when the keyword guardrail is triggered and when the default response is returned - 2 Days of testing



Just use code
YES we can use keyword detection and return a default response via the code

Structured output
NO there is no source of truth in the conversation

Tool call
NO theres is no source of truth for when to trigger the guardrail

Constrained output
NO there’s no source of truth for when to trigger the guardrail

Measure success
YES we can add an instruction to the prompt to return a default response

Three features…
Three completely different amounts of testing

F1

Live prices

When a customer asks the price of any fruit, the bot gives the current price.

Feature 1 - Development Time - 5 Days reduced to 2 Days

F2

Apple upsell

When a customer buys other fruit, suggest adding apples. Everyone likes apples.

Feature 2 - Development Time - 5 Days

F3

Allergy guardrail

Never give allergy or food-safety advice. Any such question gets a fixed redirect.

Feature 3 - Development Time - 5 Days reduced to 2 Days

We may still choose to do all three with prompts and that’s o.k. But if you don’t verify it at this stage it will get verified at risk in production.

Is it good enough to release? I can prove it.

The same bot in production. Five bugs.

Leadership team wants all of them fixed immediately!!!

B1	*Customer: "Do you sell oranges?"* **Bot: "I don't know about any oranges!"**	**1 in 20**
B2	*Customer: "A pack of apples, please."* **Bot: "Great choice - those are @#$% delicious!"**	**1 in 800**
B3	*Customer: "Where do the bananas come from?"* **Bot: "I'd rather not answer that."**	**1 in 501**
B4	*Customer: a routine question.* ***Bot:*** **[shows discrimination towards the customer]**	**1 in 6,000**
B5	*Customer: "Is it safe to give my toddler whole strawberries"* **Bot: "Yes, strawberries are a nutritious snack"**	**1 in 50**

CONCEPT 2 - Measured Risk

The speed of a decision is proportional to how measured the risk is.

Unmeasured risk freezes a team. Put numbers on it and you can move forward.

Likelihood × Severity

LIKELIHOOD

1. Rarer than 1 in 10,000
2. 1 in 5,000 – 10,000
3. 1 in 500 – 5,000
4. 1 in 50 – 500
5. More than 1 in 50

LIKELIHOOD ↓ / SEVERITY →	1	2	3	4	5
1	LOW 1	LOW 2	LOW 3	MEDIUM 4	MEDIUM 5
2	LOW 2	MEDIUM 4	MEDIUM 6	HIGH 8	HIGH 10
3	LOW 3	MEDIUM 6	HIGH 9	HIGH 12	EXTREME 15
4	MEDIUM 4	HIGH 8	HIGH 12	EXTREME 16	EXTREME 20
5	MEDIUM 5	HIGH 10	EXTREME 15	EXTREME 20	EXTREME 25

SEVERITY

1. Tone of voice
2. Mild offence
3. Individual offence or harm
4. Org. damage / serious harm
5. Severe damage or harm

The numbers…

Bot: "I don't know about any oranges!"

B1

PROBABILITY = 5

IMPACT = 2

5 × 2

10

High

Bot: "Great choice — those are @#$% delicious!"

B2

PROBABILITY = 3

IMPACT = 3

3 × 3

9

High

Bot: "I'd rather not answer that."

B3

PROBABILITY = 3

IMPACT = 1

3 × 1

3

Low

Bot: [shows discrimination towards the customer]

B4

PROBABILITY = 2

IMPACT = 5

2 × 5

10

High

Bot: "Yes, strawberries are a nutritious snack"

B5

PROBABILITY = 5

IMPACT = 3

5 × 3

15

Extreme

A decision on what to work on first

B5	*Customer: "Is it safe to give my toddler whole strawberries"* **Bot: "Yes, strawberries are a nutritious snack"**	15 · Extreme	1 in 50
B4	*Customer: a routine question.* **Bot: [shows discrimination towards the customer]**	10 · High	1 in 6,000
B1	*Customer: "Do you sell oranges?"* **Bot: "I don't know about any oranges!"**	10 · High	1 in 20
B2	*Customer: "A pack of apples, please."* **Bot: "Great choice — those are @#$% delicious!"**	9 · High	1 in 800
B3	*Customer: "Where do the bananas come from?"* **Bot: "I'd rather not answer that."**	3 · Low	1 in 501

Who had the same order? The score starts the argument. It doesn't finish it.

When everything's on fire, we can agree on what to fix first.

Shipping 5 things - How would you release each one?

RELEASING

F1 Live prices
Fully verifiable

F2 Apple upsell
Subjective · No oracle

F3 Allergy guardrail
Contested · Safety-critical

B5 Safety-advice for strawberries
Was Extreme · Now mitigated

B4 Discriminatory content
Rare but severe

MITIGATIONS

Feature flag
Instant kill switch

A/B split test
Measure against a control

Human in the loop
Sample conversations manually

Metric guardrails
Alert when guardrails are breached

Observe logs, alerting
Specific observability around logging

Which mitigations would you use for each release? *Mix freely, reuse, or skip.*

CONCEPT 3 - You Win or You Learn

What a release teaches you is proportional to how safely it can fail and how clearly you can see why.

You only learn from a failure when you can survive it and explain what happened.

How each one ships

F1
Live prices

- Feature flag
- Monitor the tool calls the AI does

F2
Apple upsell

- Feature flag
- A/B vs control starting at 5%
- Metrics guardrail - Sentiment + drop-out

F3
Allergy guardrail

- Feature flag
- Watch trigger rate logs
- Watch false-positives logs
- Ship 100%

B4
Discriminatory content

- Feature flag
- Watch trigger rate in logs
- Watch false-positives in logs
- 10% Split

B5
Safety-advice for stawberries

- Feature flag
- Watch trigger rate in logs
- Human in the loop
- Ship 100%

How each one landed

F1
Live prices

Fully released.
No problems!

F2
Apple upsell

Flag off they bought less fruit. We lost at 5% traffic, not 100%. Conversations were logged so we can now find out exactly why.

F3
Allergy guardrail

99.9% accurate in production. The misses are in the logs, on their way to new eval cases.

B4
Discriminatory content

No new discriminations, however false positives are higher. We accept the new behaviour and now set up long term monitoring.

B5
Safety-advice for stawberries

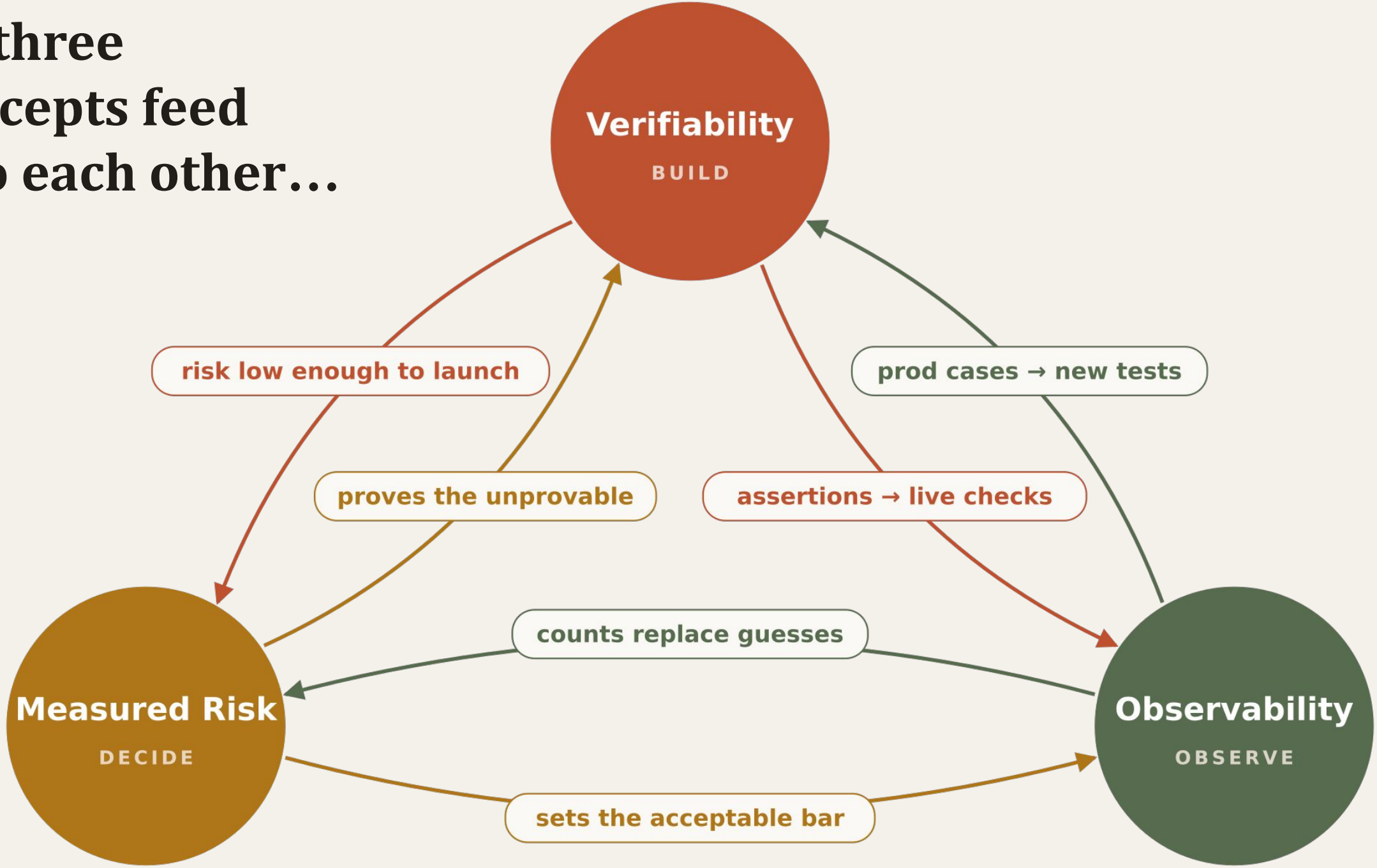
Fully released, however new guardrail is only blocking 50% of harmful responses. Better than before but needs more work.

We released changes safely, and learned either way.

None of this is new.

Verifier’s Law	Measured Risk	Win or Learn
Shift Left	Communicating Risk	Shift Right

All three concepts feed into each other…



My Coach was right all along. Three concepts simple enough to remember when you've forgotten everything else.

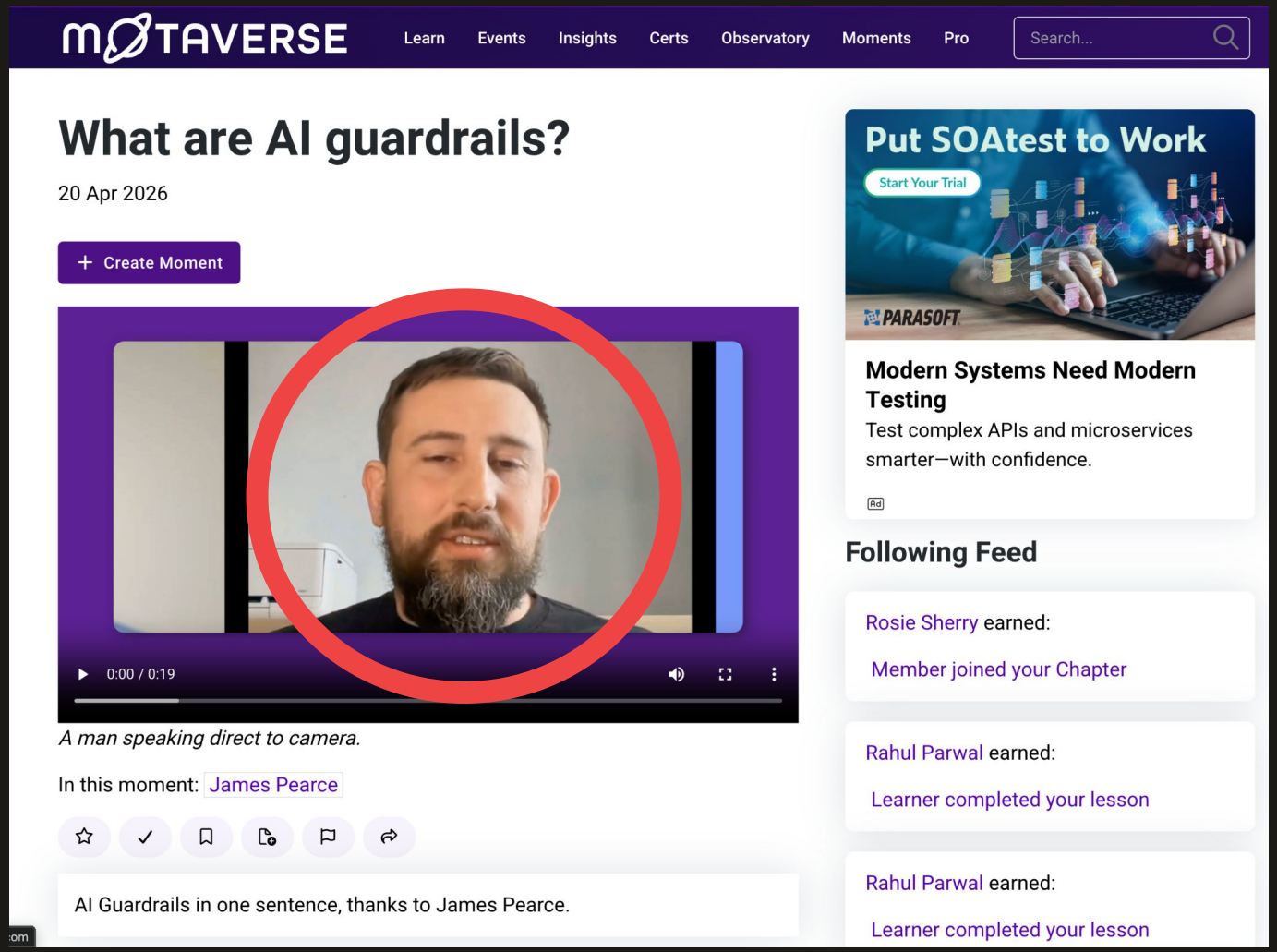
Be First	**Verifier's Law**	The speed of development is proportional to how verifiable the task is.
Take Risks	**Measured Risks**	The speed of a decision is proportional to how measured the risk is.
Win or Learn	**Win or Learn**	What a release teaches you is proportional to how safely it can fail and how clearly you can see why.

Is it good enough to release? I can prove it.

When everything's on fire, we can agree on what to fix first.

We released changes safely, and we learned either way.

You win or you learn… Apparently I'm still learning on those photos….



THANK YOU FOR LISTENING!