Building AI That Ops Teams Actually Trust

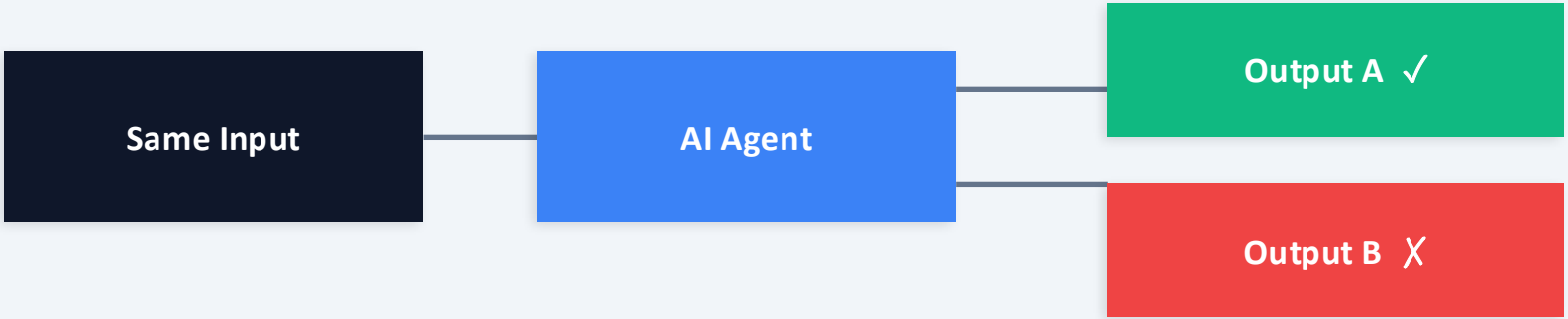
Lightning Talk - PyTorch Conference Europe 2026

Robert King · Palo Alto Networks · Lead Solutions Engineer
linkedin.com/in/kingwebstring/

Prove it.

The default response from every ops engineer when you show up with an AI agent.

The Non-Determinism Problem



Traditional testing assumes determinism.

That breaks down with AI agents.

Testing Like a Chess Coach

Opening Preparation

Tactical Calculation

Positional Evaluation

Endgame Technique

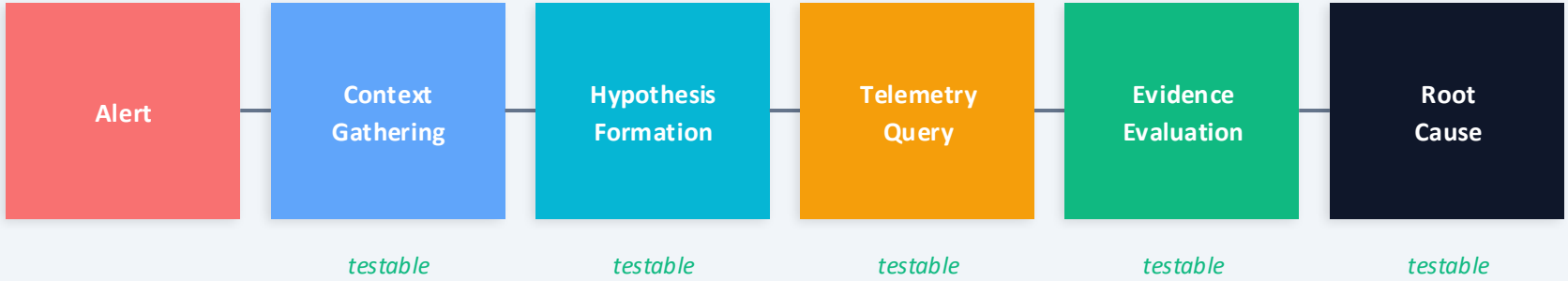
Each phase is testable independently

When a player loses, you don't just say *'they played badly.'*

You pinpoint which phase went wrong.

Same principle: decompose to diagnose

Modular Sub-Agent Architecture



Each sub-agent handles a discrete step.

When something breaks, you know exactly which component failed.

A Real Validation Scenario

1. **Detect error rate spike**
   Pattern recognition
2. **Identify downstream service**
   Trace analysis
3. **Navigate to service, see crash loop**
   Service navigation
4. **Notice increased error logs**
   Log analysis
5. **Connect to feature flag flip**
   Root cause

Outcome vs Trajectory Evaluation

Outcome Evaluation

Did the agent get the final answer right?

PASS / FAIL

Coarse — no insight into why

VS

Trajectory Evaluation

Did the agent take the right steps to get there?

Granular — precision, recall, progress rate

Right Answer. Wrong Path.

1. Detect error rate spike
2. Analyse service dependencies
3. Navigate to service — *SKIPPED*
4. Notice error logs
5. Feature flag flip = root cause

Answer: Correct ✓

Test: FAILED ✗

If the engineer can't verify the reasoning, they won't trust it.

Three Layers of Evaluation

Operations

Did each individual operation do the right thing?

Correct operation? Right API called?

End-to-end

Did the full execution path make sense?

Valid reasoning chain? No unnecessary detours?

Output

Did the agent actually resolve the problem?

Correct root cause? Complete investigation?

An output pass with an end-to-end fail = unsound sacrifice.

It worked once. It won't work next time.

Synthetic vs Real-World Test Data

Synthetic	Real World
Clean latency spike ↓ Obvious database bottleneck ↓ Single root cause	Latency spike + deployment + config change + unrelated upstream errors ↓ Signal buried in noise

Root causes we test against:

- Feature flag flips
- Bad deploys
- OOMs
- Upstream failures

Validation at Scale

2,600+

investigations per week
in production

600+

tests including
hundreds of golden trajectories

Complete multi-step reasoning per investigation

Minutes

to detect regressions
not weeks

Earning Trust

Show Your Reasoning

Data considered, hypotheses evaluated, alternatives ruled out.
Not a confidence score.

Shadow Mode

Agent runs alongside engineers.
Not in the critical path.
They observe, they verify.

Earned Trust

Engineers check it earlier.
Adoption grows from observation, not mandate.

The Framework

Build modularly - evaluate trajectories, not just outcomes

Test at operation, end-to-end, and output level

Ground tests in real-world failure modes

Show your reasoning - the full chain

Earn trust gradually - observation, not mandate

Robert King · Palo Alto Networks · Lead Solutions Engineer - linkedin.com/in/kingwebstring/