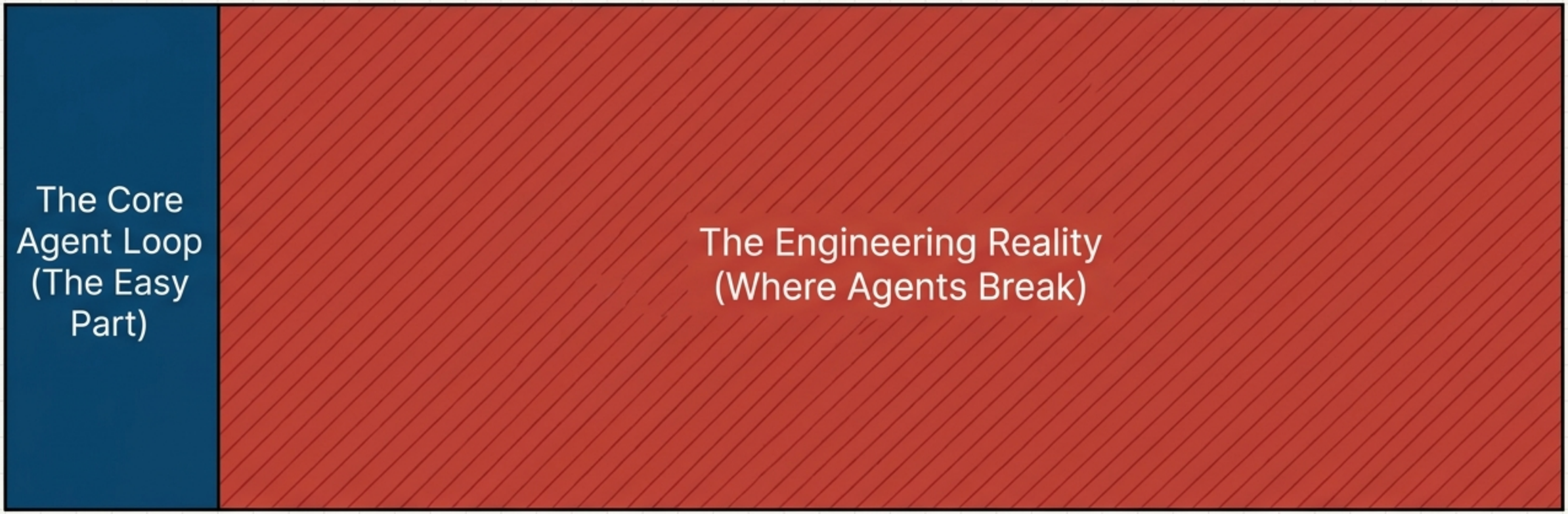
# The 90% Problem: Why Most AI Agents Are Still Broken




Building an agent that works is a weekend project. Building one that doesn't break is 90% of the work.

# The Five Pillars of Agent Architecture

| Pillar | What It Does | Who Does the Work |
| --- | --- | --- |
| **Tool Use** | Interacts with the world | LLM-dominated |
| **Planning** | Sequences steps | LLM-dominated |
| **Reflection** | Judges output | Mixed |
| **Memory** | Cross-session persistence | Code-dominated |
| **Context** | Prompt state management | Code-dominated |

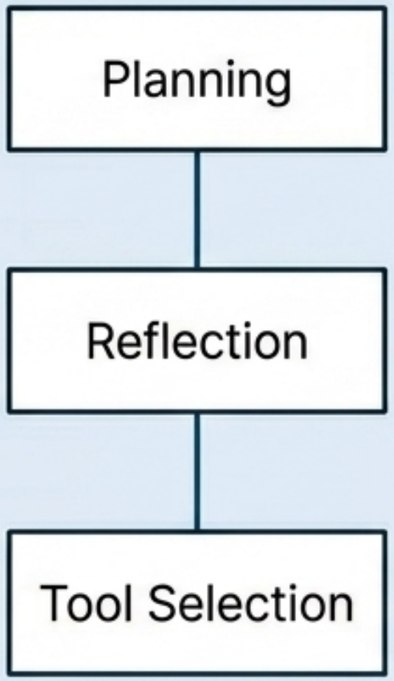
Free upgrades via model improvements (e.g., GPT-4 to Claude Opus).

**Engineering problems.** A model upgrade won't fix a pipeline that dumps 10 irrelevant entities into a prompt.

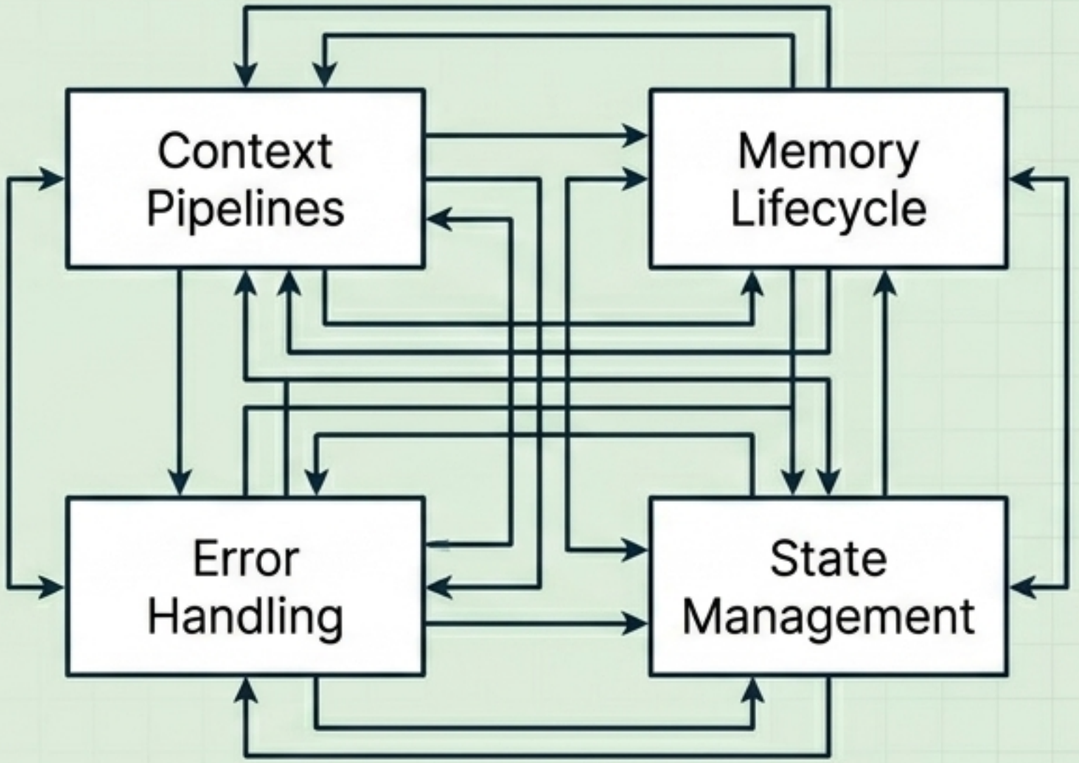
Where production agents actually fail.

# The LLM vs. Code Divide

## LLM Drives




## Code Drives




Low LLM dependence ≠ Zero LLM calls. The failure mode is the orchestration.

# Three Ways Agents Call LLMs

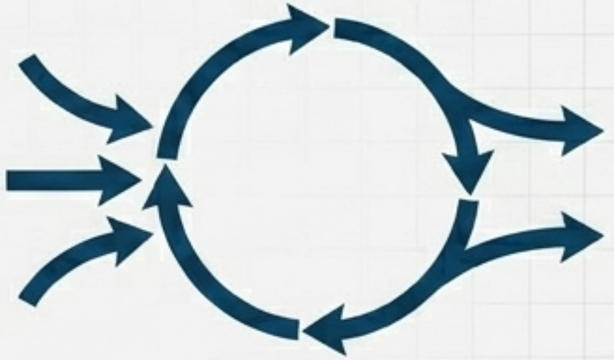
## sideQuery

| | |
|---|---|
| **Metrics:** | 256 tokens / ~250ms |
| **Behavior:** | Code asks, code acts. |
| **Example:** | (Semantic retrieval from 200 files) |

## Fork Agent

| | |
|---|---|
| **Metrics:** | 10K+ tokens |
| **Behavior:** | Code delegates, agent delivers. |
| **Example:** | (Autocompact summarization) |

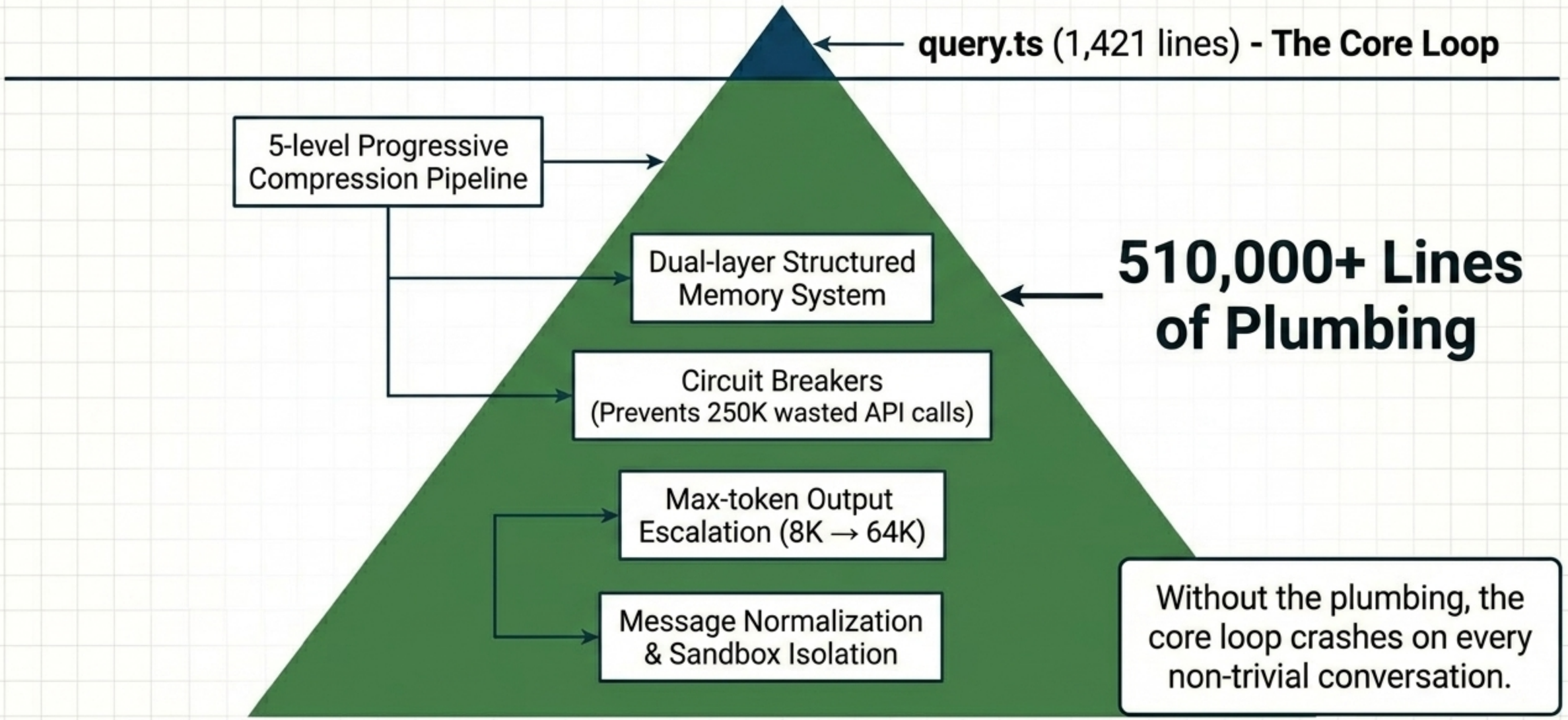
## Main Tool Loop



| | |
|---|---|
| **Metrics:** | Infinite turns |
| **Behavior:** | LLM triggers, code executes. |
| **Example:** | (The core agent orchestrator) |

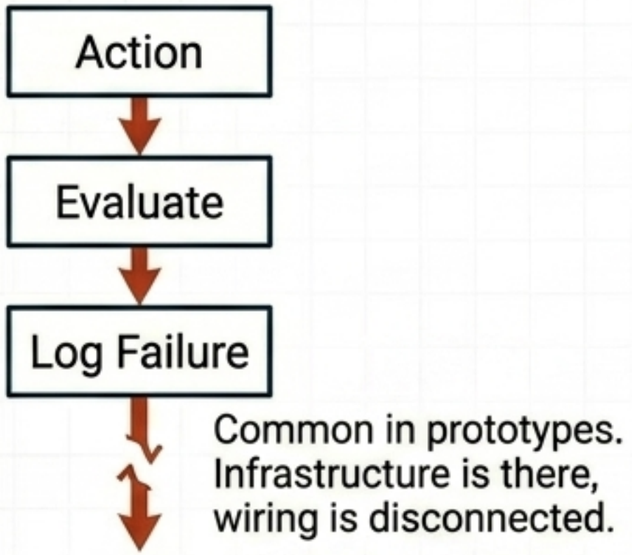
Choosing the wrong pattern wastes budget and destroys reliability.

# The Anatomy of Production: Claude Code

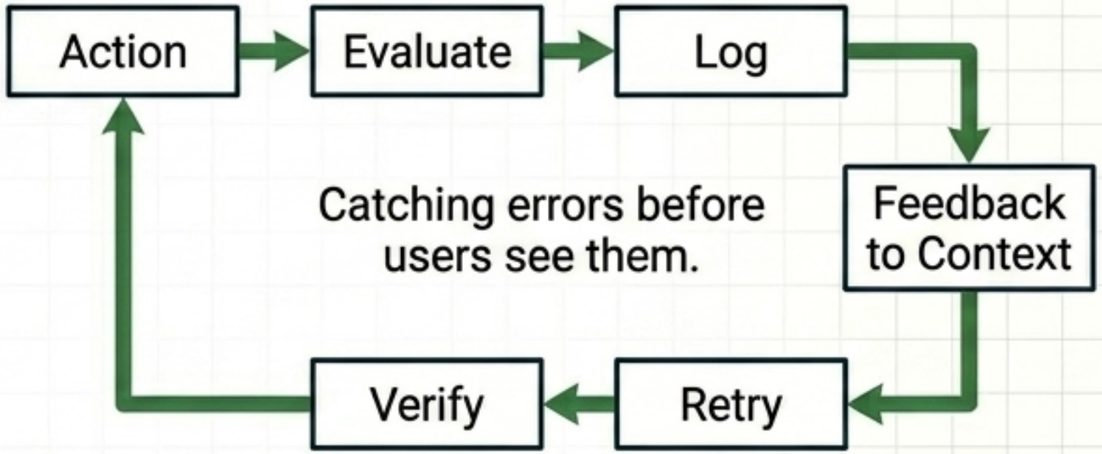


# “Features Exist” vs. “Loops Are Closed”

## Features Exist (Open Loop)



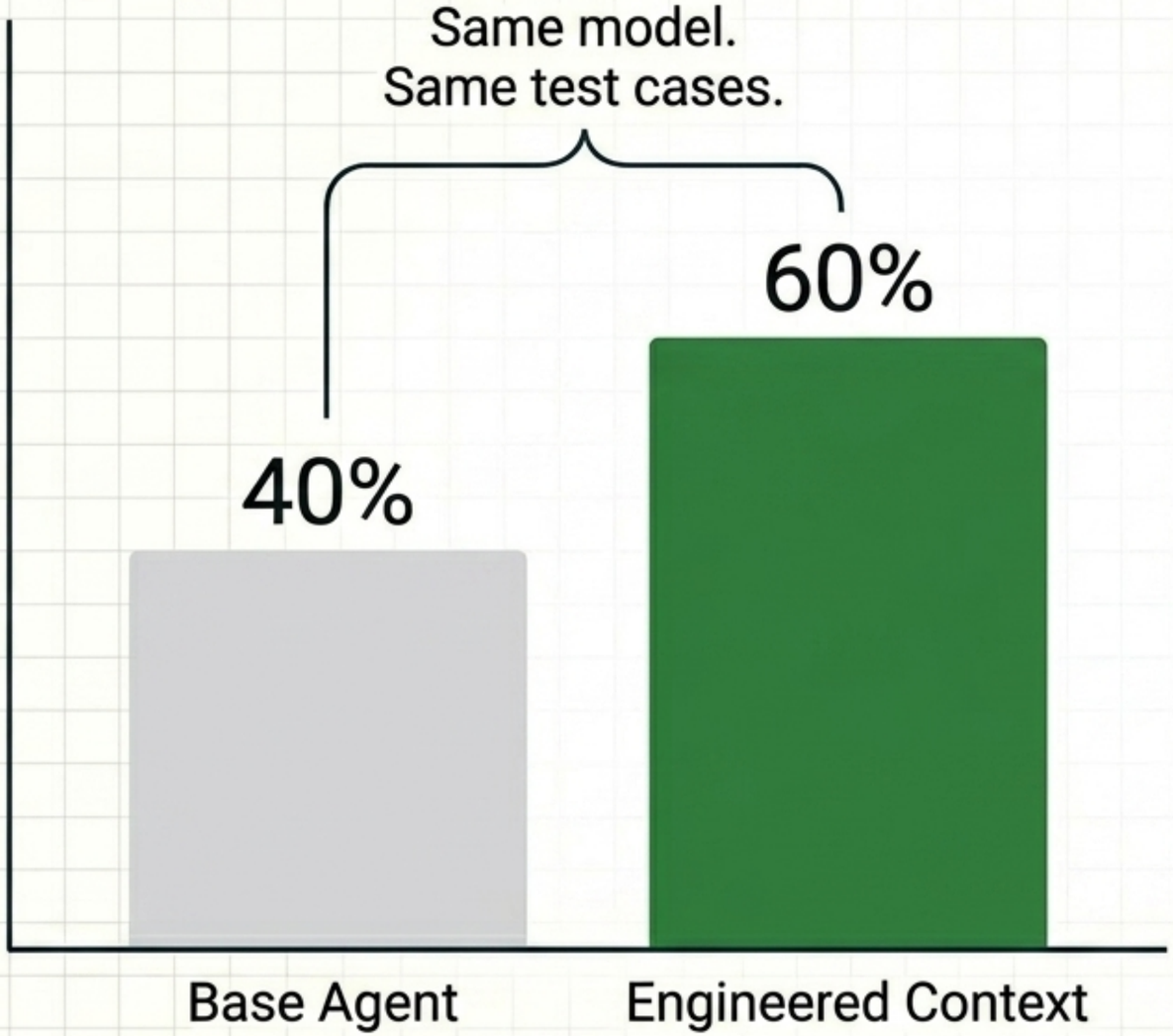

## Loops Are Closed




| Practical Examples | | |
|---|---|---|
| **Verification** | Log only | Retry on failure |
| **Memory** | Store indefinitely | Decay lifecycle |
| **Compression** | Truncate randomly | Circuit break |

Half-built infrastructure is worse than none—it gives false confidence.

# The Proof: Code Over Capability




## 7 out of 8 fixes = Pure Code. Zero LLM cost.

- Priority bounding
- Structured error classification
- Truncation conclusions
- Circuit breakers

The model was always capable.
The context was suffocating it.

# The model is a commodity.

## The engineering around it is the product.