上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

Lecture 36: Long-Horizon Reasoning

Tao Huang

John Hopcroft Center, School of Computer Science, Shanghai Jiao Tong University

https://taohuang.info/cs3317

https://oc.sjtu.edu.cn/courses/89538

AI tools assisted in generating some figures in these slides. All such content has been reviewed, and the instructor is responsible for its accuracy.

From a Single Step to Fifty

L35 gave the agent a loop. But the loop falls apart over long horizons.

- **L35 recap:** the ReAct loop — think → act → observe. One step works. Your **coding agent** ran read → edit → test.
- **The catch:** a real task isn't one step. It's **fifty** — "refactor this module," "research this question."
- **Today:** what breaks when we chain steps, and how we patch it.

The Central Puzzle

A single step works 95% of the time.
So why does a 50-step task fail 90% of the time?

Two culprits, all lecture:

① the reasoning **drifts**

② the memory **runs out**

Objectives

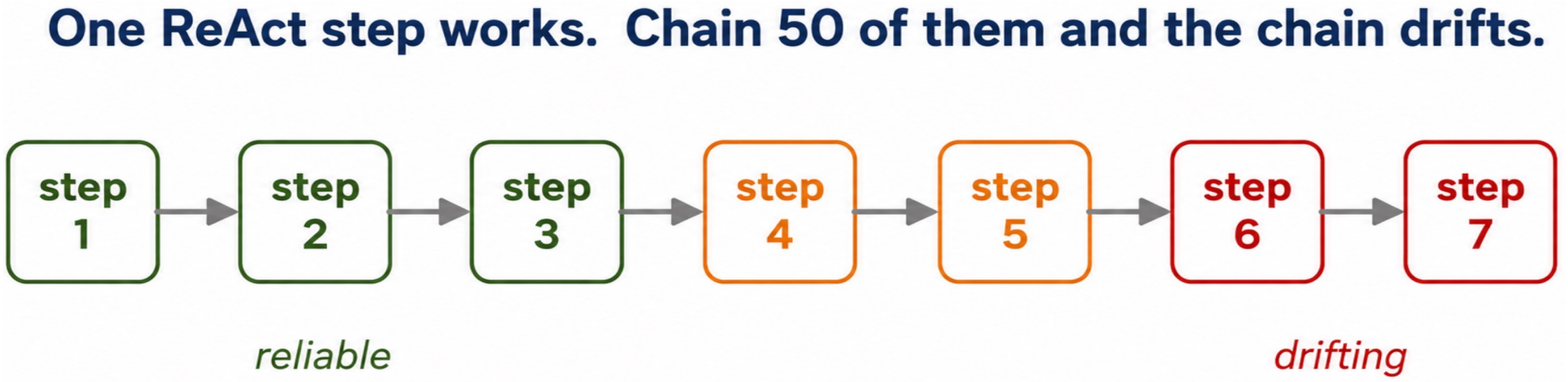
By the end of this lecture, you will be able to

- **Explain** why long tasks fail even when each step is reliable (error compounding).
- **Distinguish** the two failure modes — reasoning drift vs. memory overflow.
- **Compare** decomposition, reflection, and search as reasoning strategies.
- **Sketch** an agent's memory hierarchy: working → summarized → long-term.
- **Evaluate** which technique fixes which failure mode.

1. Why the Loop Breaks

From One Step to Many

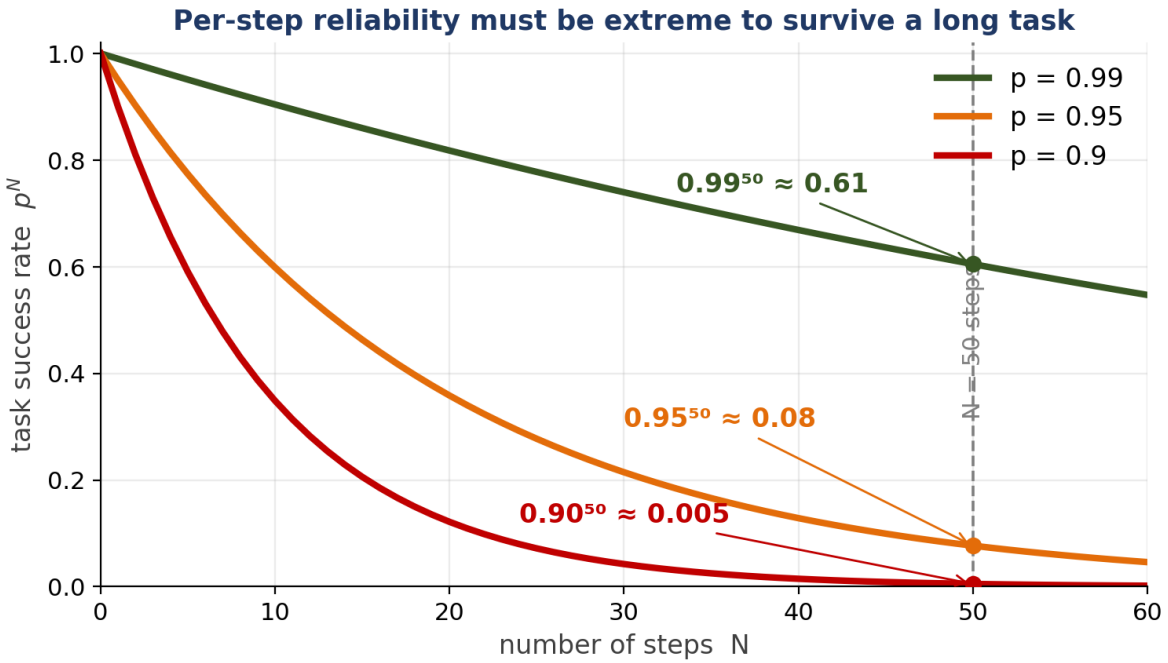
- **Situation:** the ReAct loop handles one think → act → observe cycle well.
- **Complication:** a real task is dozens of those cycles, chained back-to-back.
- **Question:** *if every step is reliable, why does the chain collapse?*



Error Compounding

If each step succeeds with probability p, an N-step task succeeds with:

$$P_{\text{task}} = p^N$$



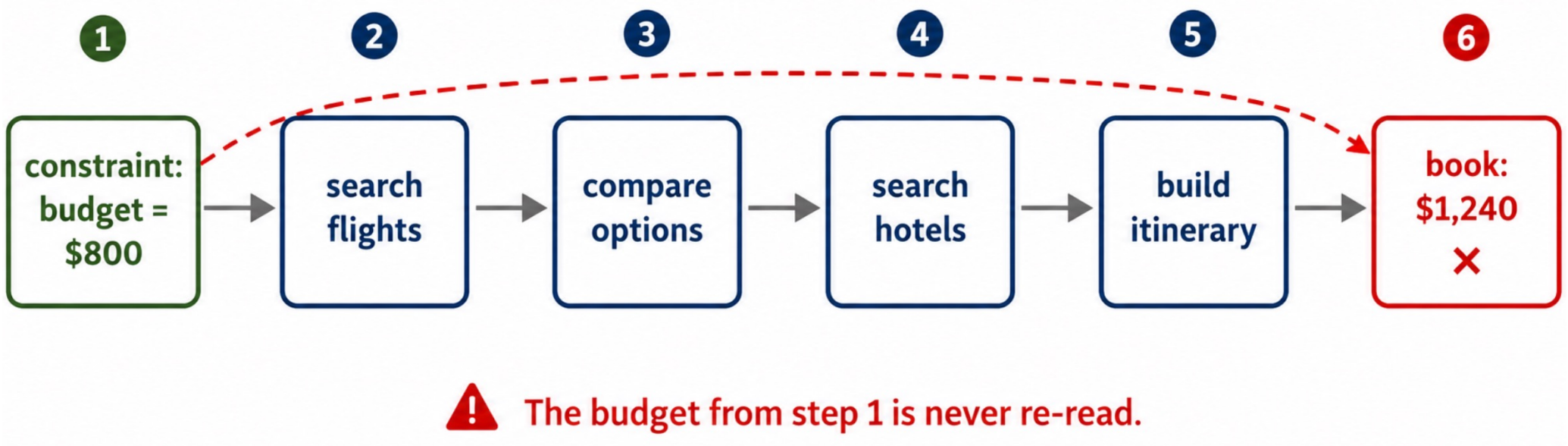
Per-step reliability must be near-perfect for a long task to work at all.

Two Ways the Chain Fails

Coding lens: re-runs the same failing test 10× · forgets it already edited a file.

Think: Where Does This Break?

This 6-step trip-booking agent ends up over budget. Which failure mode — drift or memory?



Discuss before the reveal.

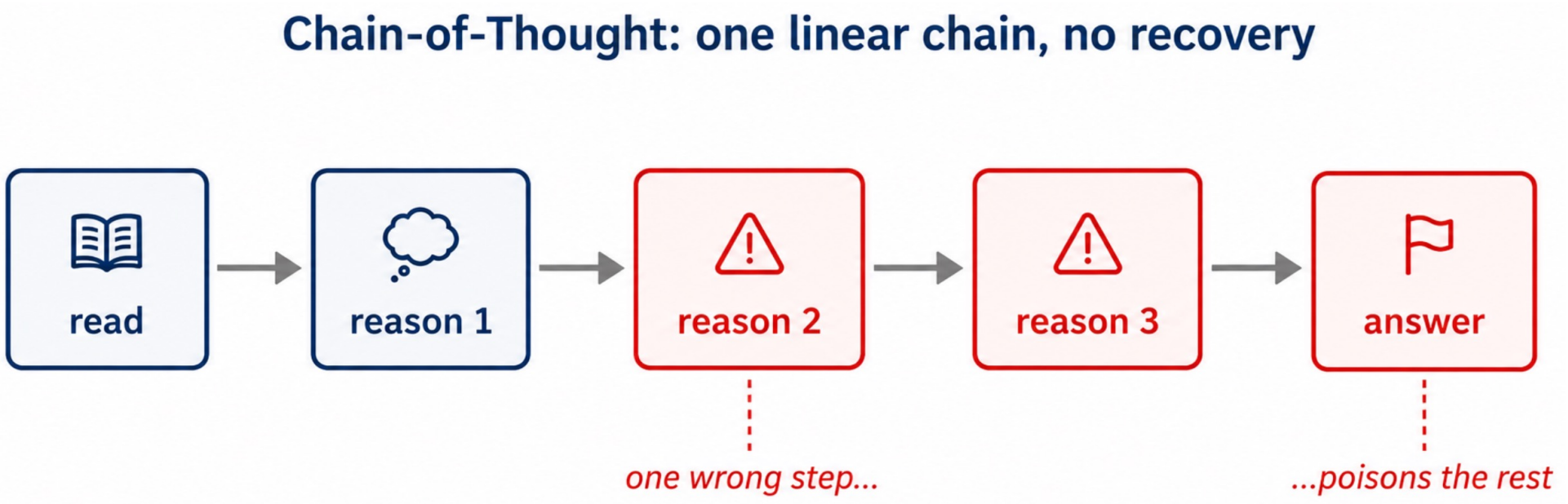
2. Reasoning Over Many Steps

Structuring the Thinking

- **Situation:** the model can already "think out loud" — Chain-of-Thought.
- **Complication:** one straight chain can't hold a 50-step task — no plan, no recovery, no backup path.
- **Question:** *how do we structure reasoning so it survives many steps?*
- **Answer:** decompose it, reflect on it, and — when stuck — search alternatives.

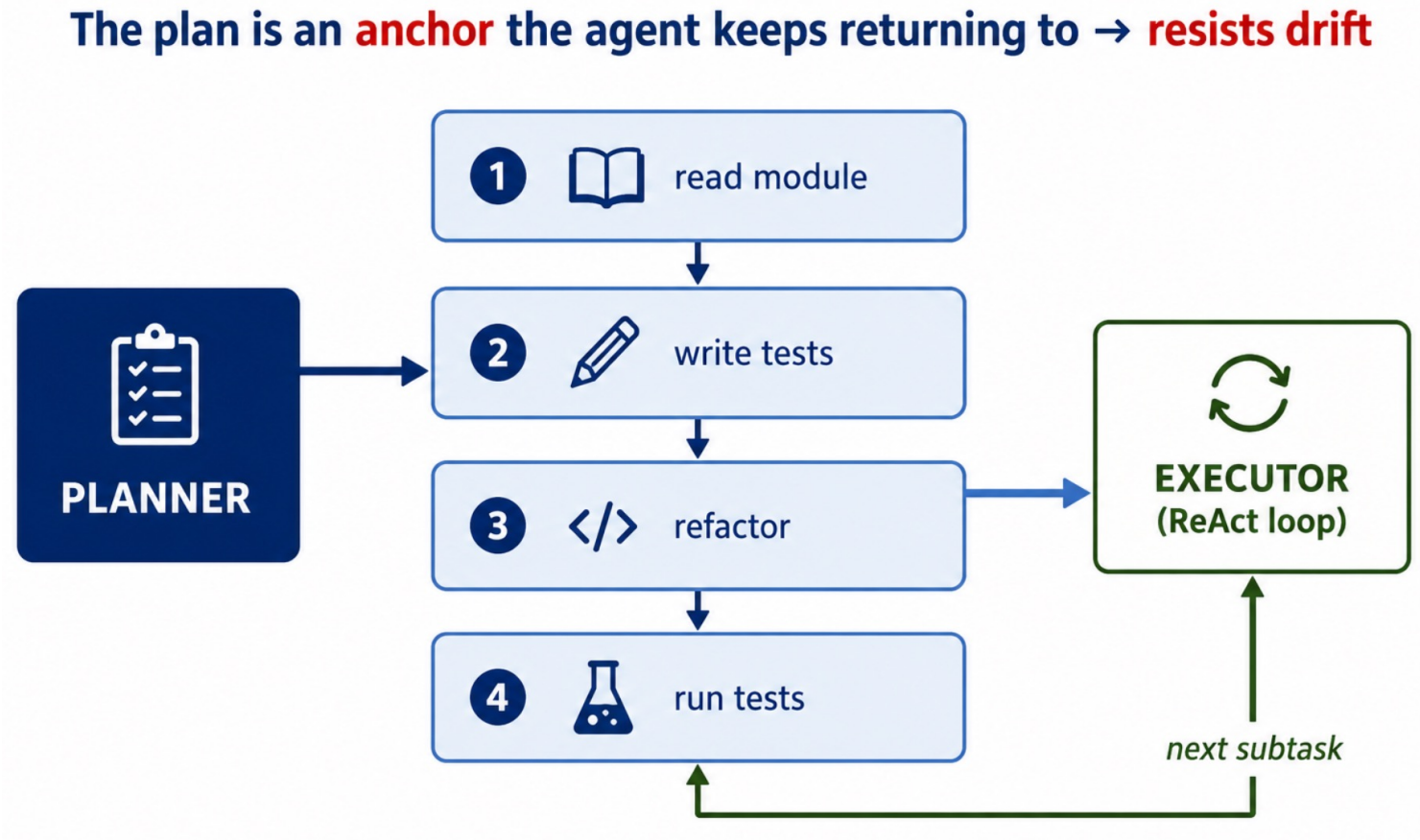
Chain-of-Thought, Briefly

- **"Let's think step by step"** — the model writes intermediate reasoning before answering.
- **Why it helps:** turns one hard leap into many small ones.
- **The limit:** it's a single linear chain — one wrong step poisons everything after it.



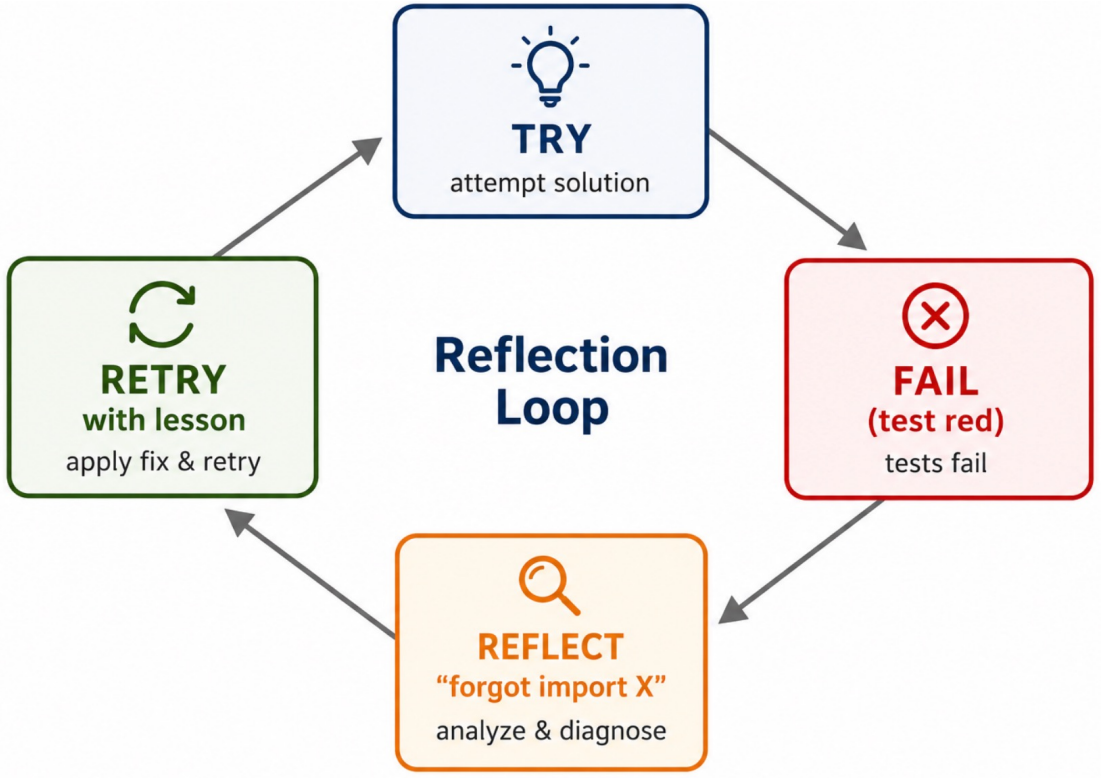
Strategy 1 — Decompose & Plan

- **Plan-and-execute:** first write a plan (sub-goals), then execute each in turn.
- The plan is an anchor — the agent keeps returning to it, which resists drift.



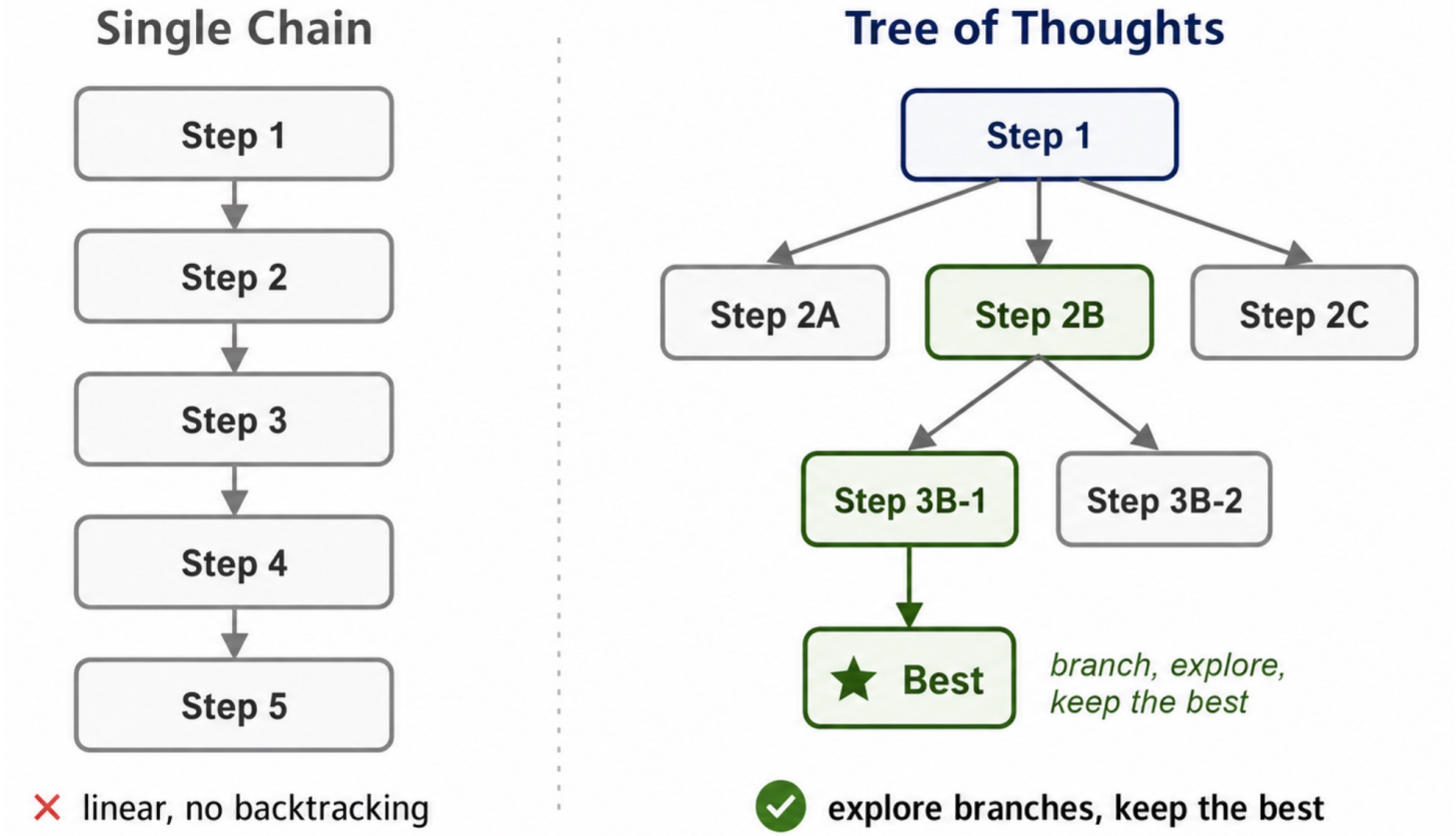
Strategy 2 — Reflect & Self-Correct

- **Reflexion idea:** try → fail → write down ***why*** it failed → retry with that lesson in hand.
- The written reflection becomes a note the agent reads on the next attempt.



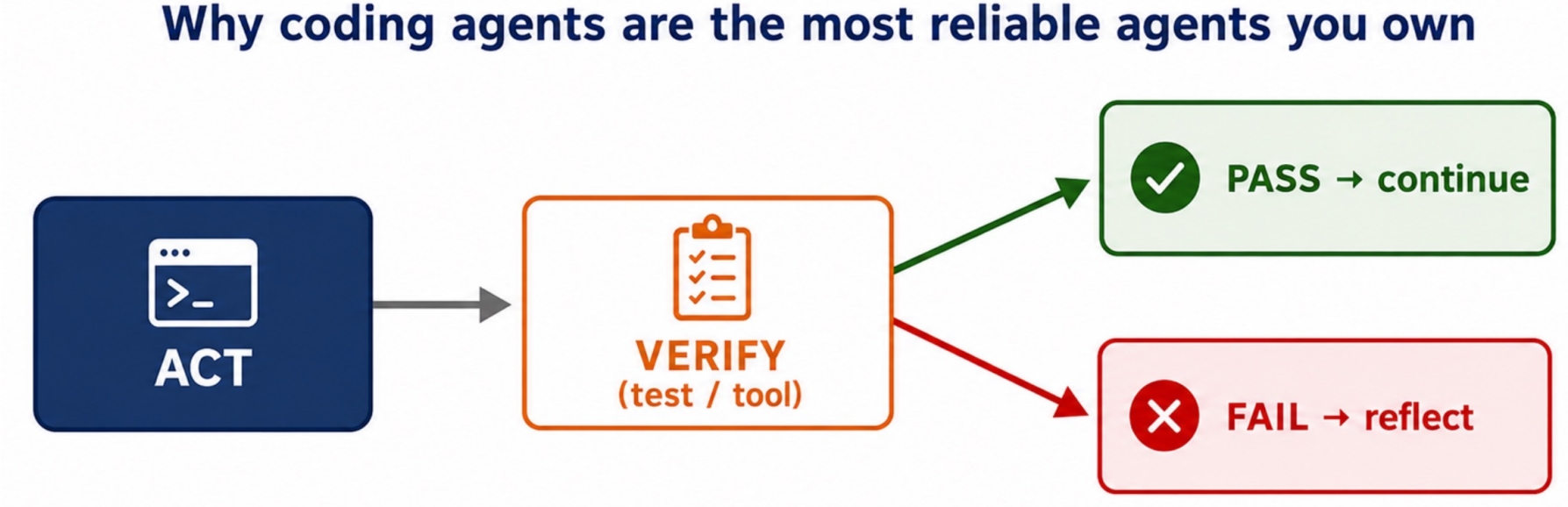
Strategy 3 — Search Alternatives

- **Tree of Thoughts:** when one path is uncertain, **branch** — explore several, keep the best.
  Trades compute for reliability. (Illustrative — we won't go deep.)



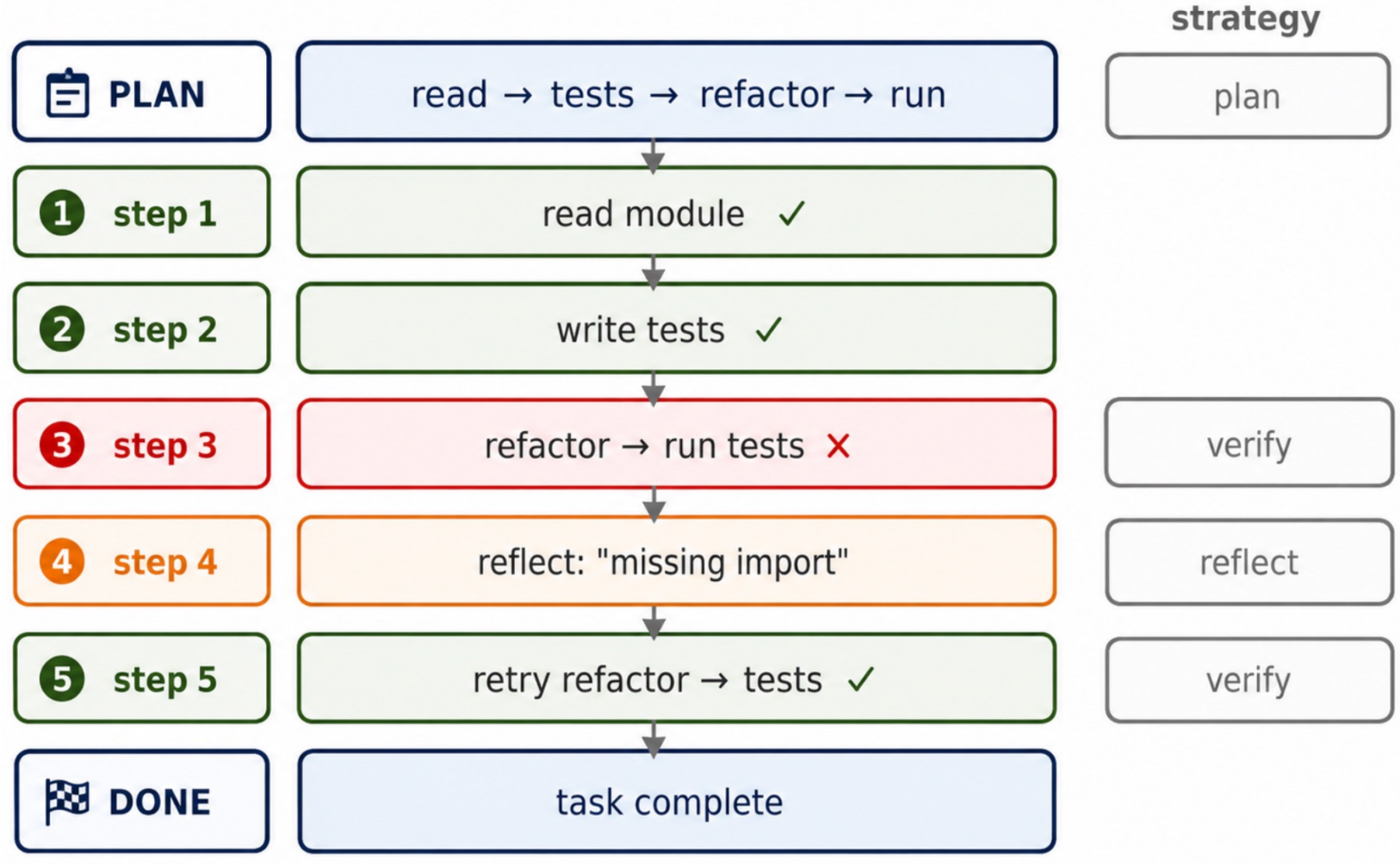
Verification — Check Your Own Work

- A separate **check** step — the model, a unit test, or a tool — validates before moving on.
- **This is why coding agents are the most reliable agents you own:** the test suite is a free, automatic verifier.



Worked Example — A Coding Agent

- **One multi-file task. Watch plan, reflect, and verify all fire:**



But All Three Assume One Thing

Plan, **reflect**, **verify** — every one assumes the agent can **remember**:

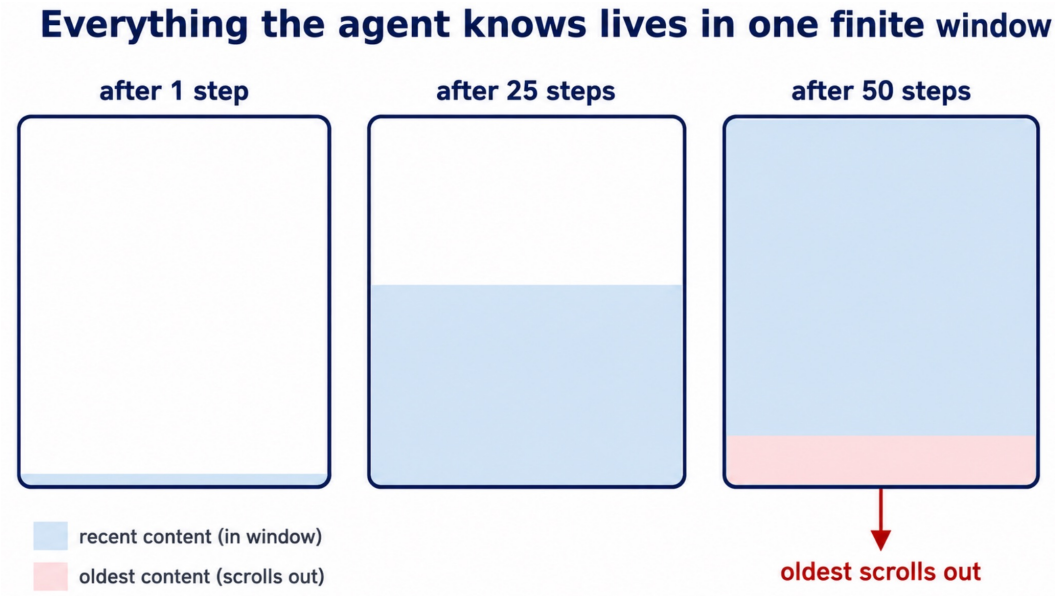
- its plan, 40 steps later
- the reflection it wrote three attempts ago
- the result of a subtask it already finished

Over 50 steps — can it? That's the memory problem.

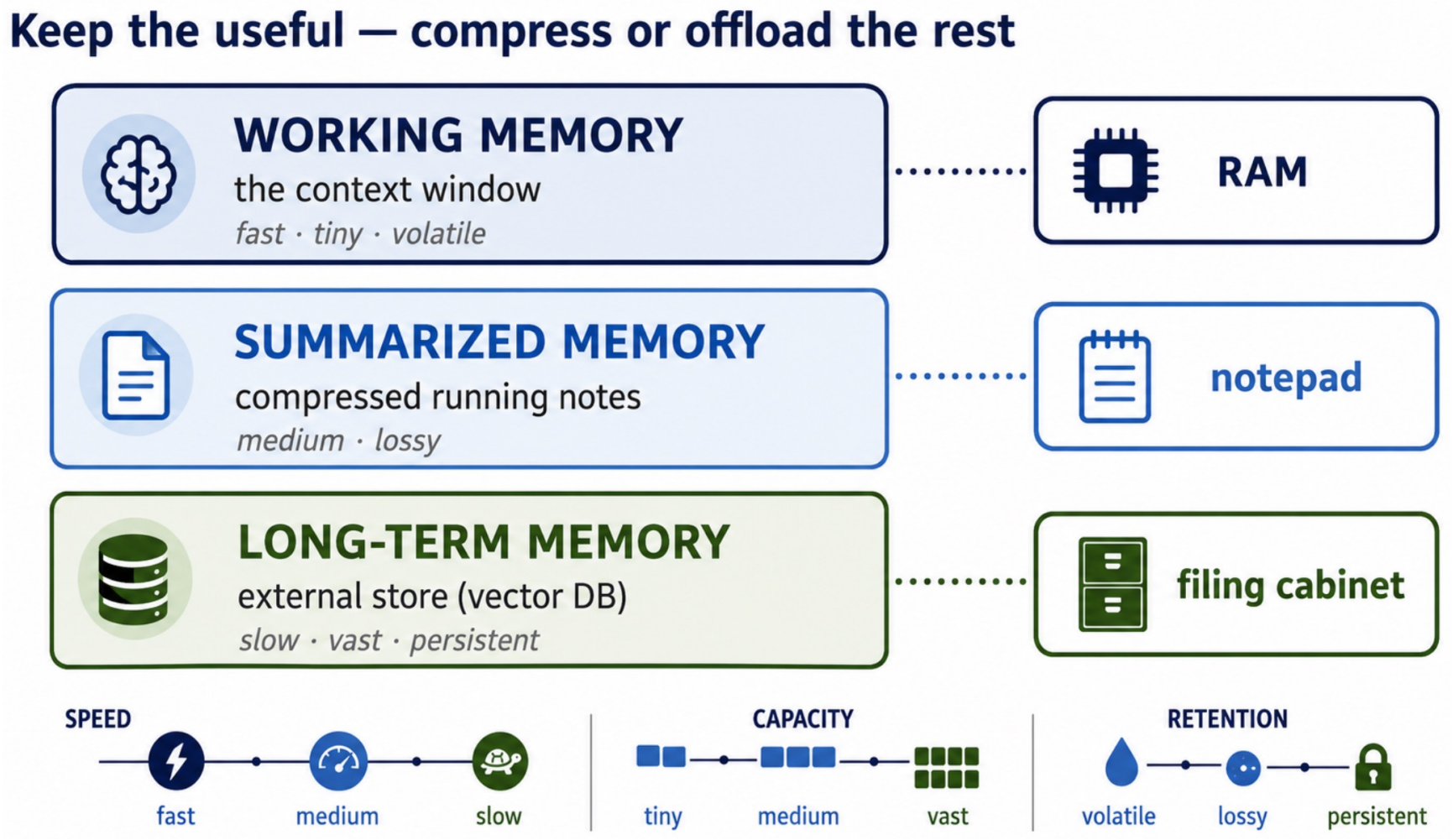
3. Memory

The Context Window Runs Out

- **Situation:** everything the agent knows lives in its context window — its working memory.
- **Complication:** 50 steps overflow even a 200K-token window, and models attend poorly to the middle ("lost in the middle").
- **Question:** *how does an agent remember beyond what fits in the window?*
- **Answer:** a memory system — keep the useful, compress or offload the rest.



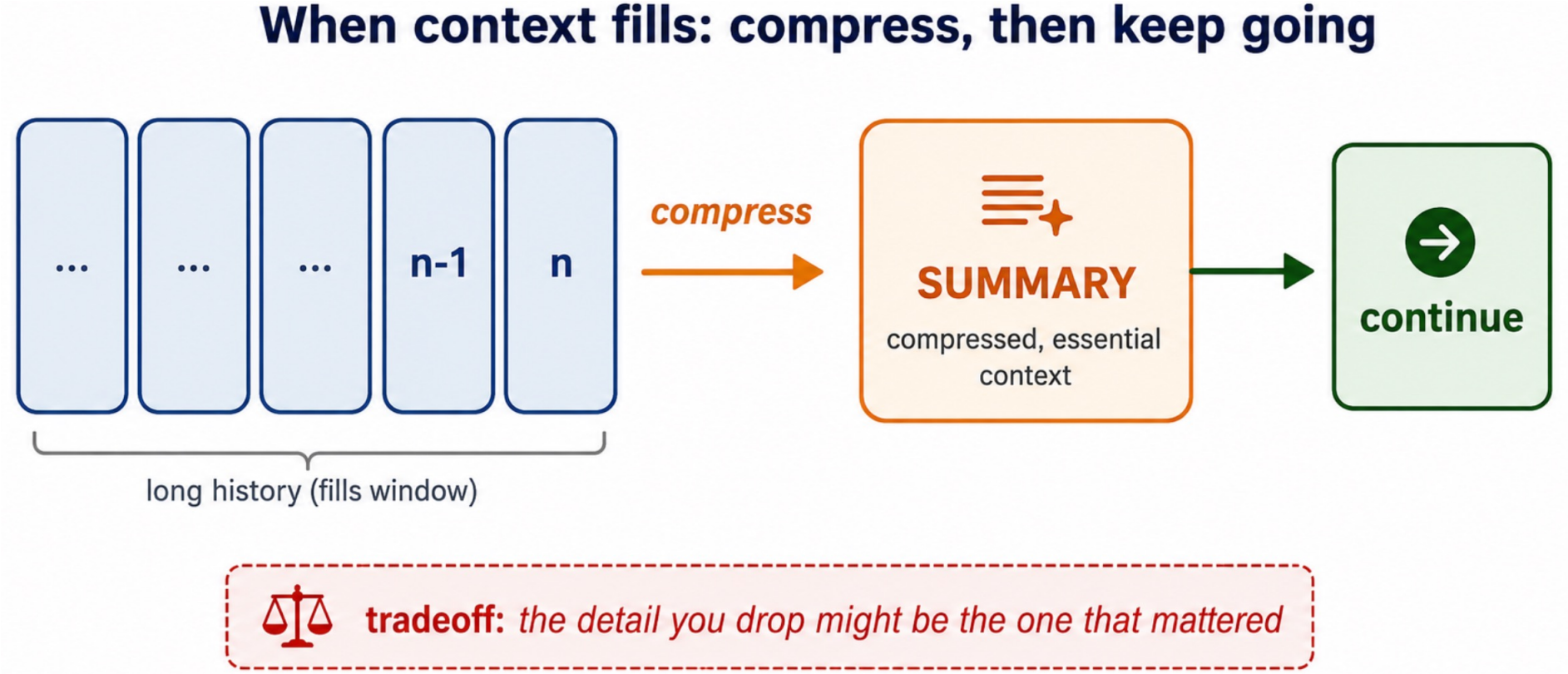
A Memory Hierarchy



Analogy: RAM vs. notepad vs. filing cabinet — RAG (L35) is how we reach the cabinet.

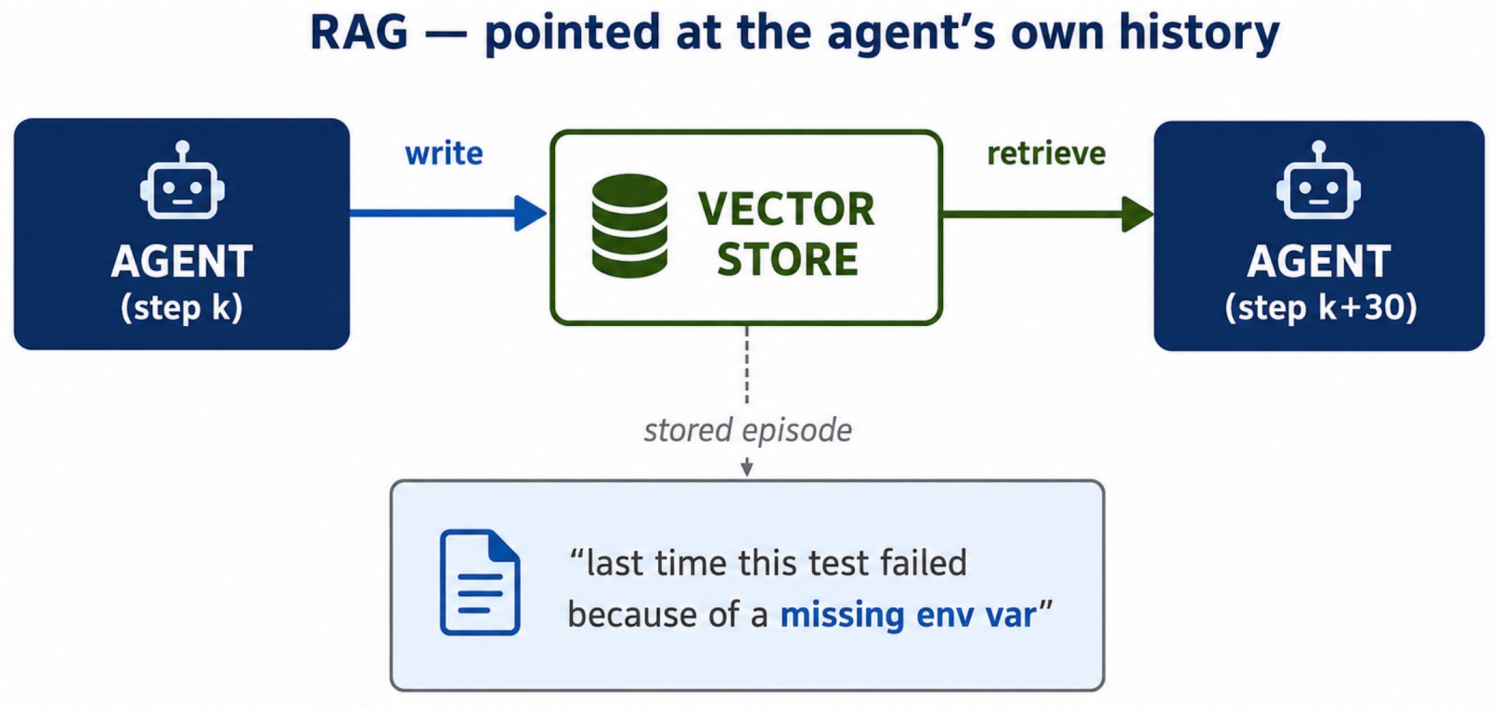
Summarization & Compaction

- **When context fills:** compress the history into a summary, then keep going.
- **Concrete:** Claude Code's "auto-compact"

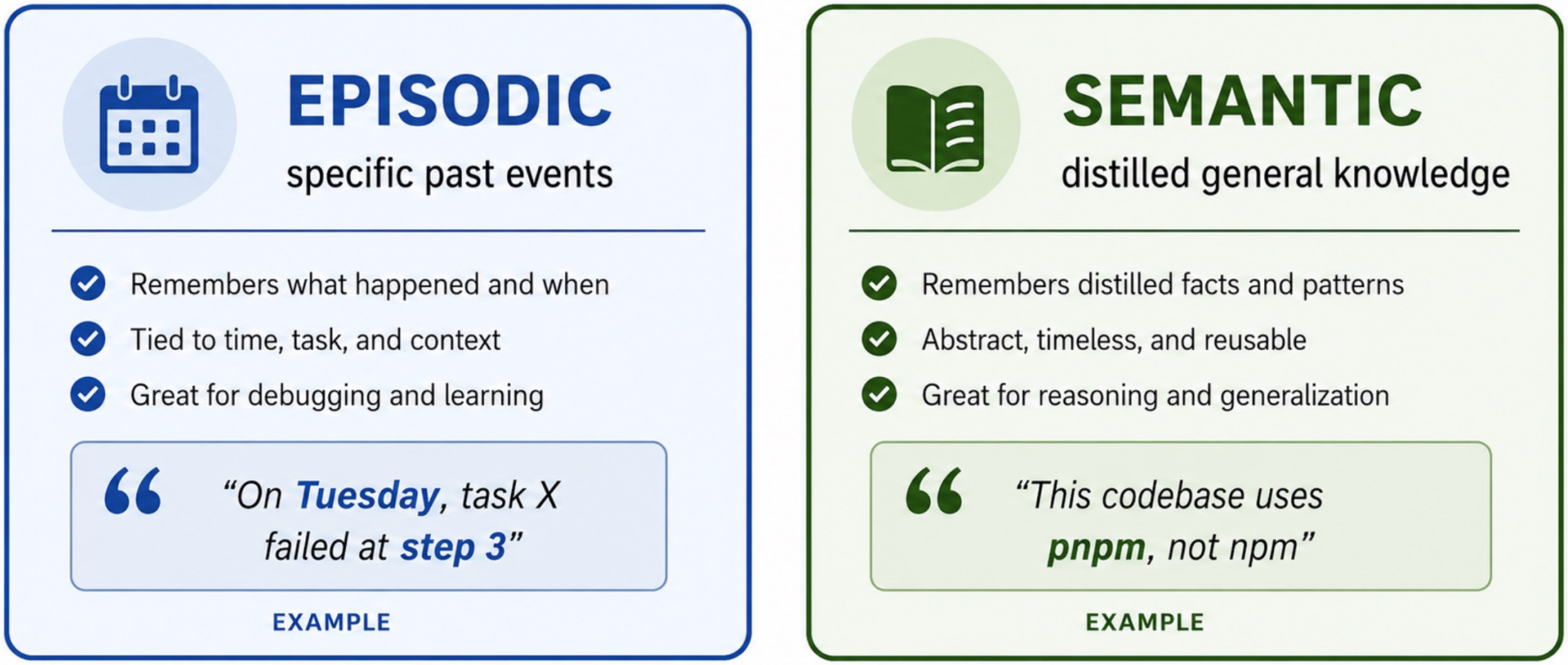


Long-Term Memory — Write & Retrieve

- Store traces and facts in an external store (a vector DB); retrieve the relevant ones on demand.
- **It's RAG — pointed at the agent's own history** (the same tool from L35, turned inward).

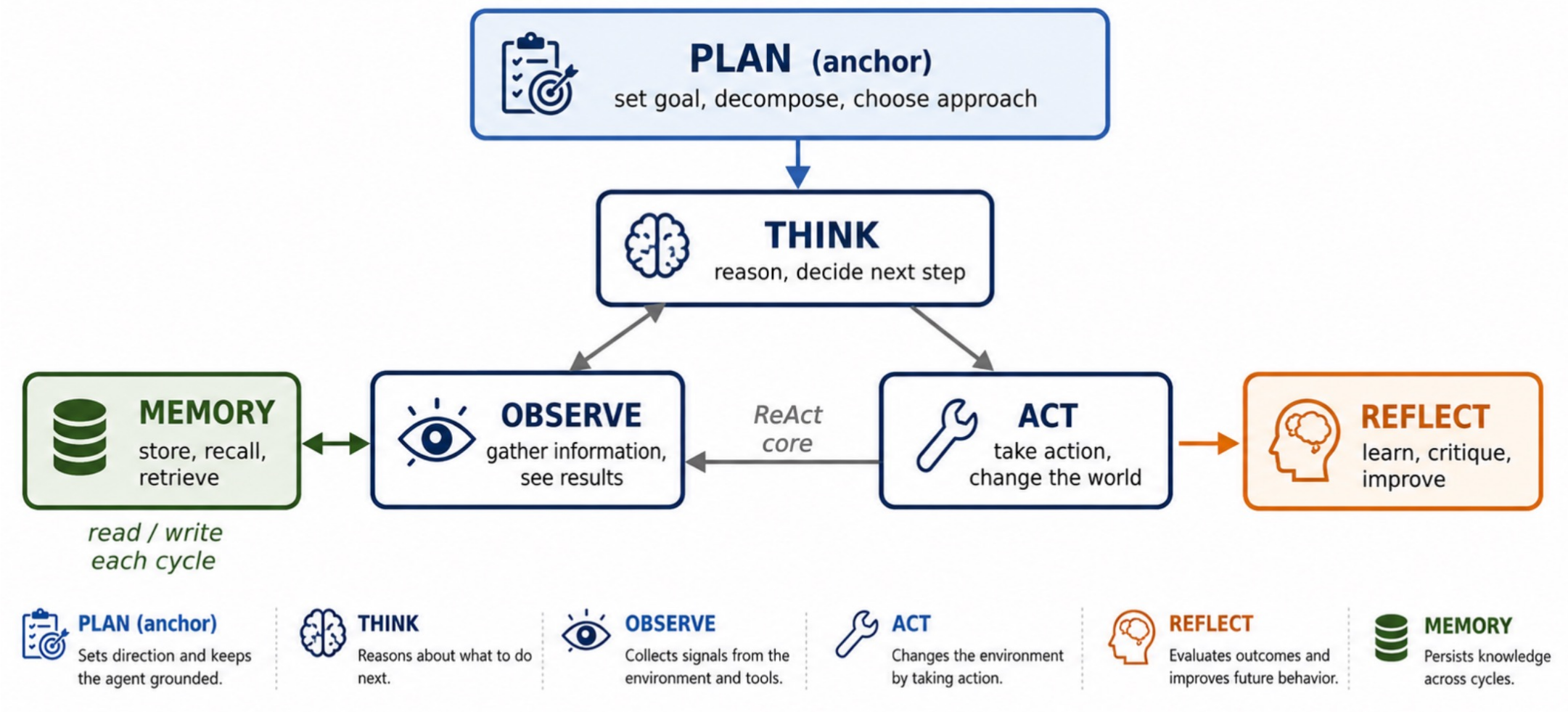


Episodic vs. Semantic Memory



- **Episodic** = what happened. **Semantic** = what's generally true.
- *An agent that keeps both stops re-learning the same lesson every session.*

Putting It Together — The Loop, Upgraded



Same ReAct core from L35 — now with a plan anchor, a reflection step, and memory I/O.

Think: Which Fix?

For each failure, pick the technique that addresses it.

Match each to: reflection · long-term memory · verification

(a)

stuck repeating the same action

(b)

forgot the budget from step 1

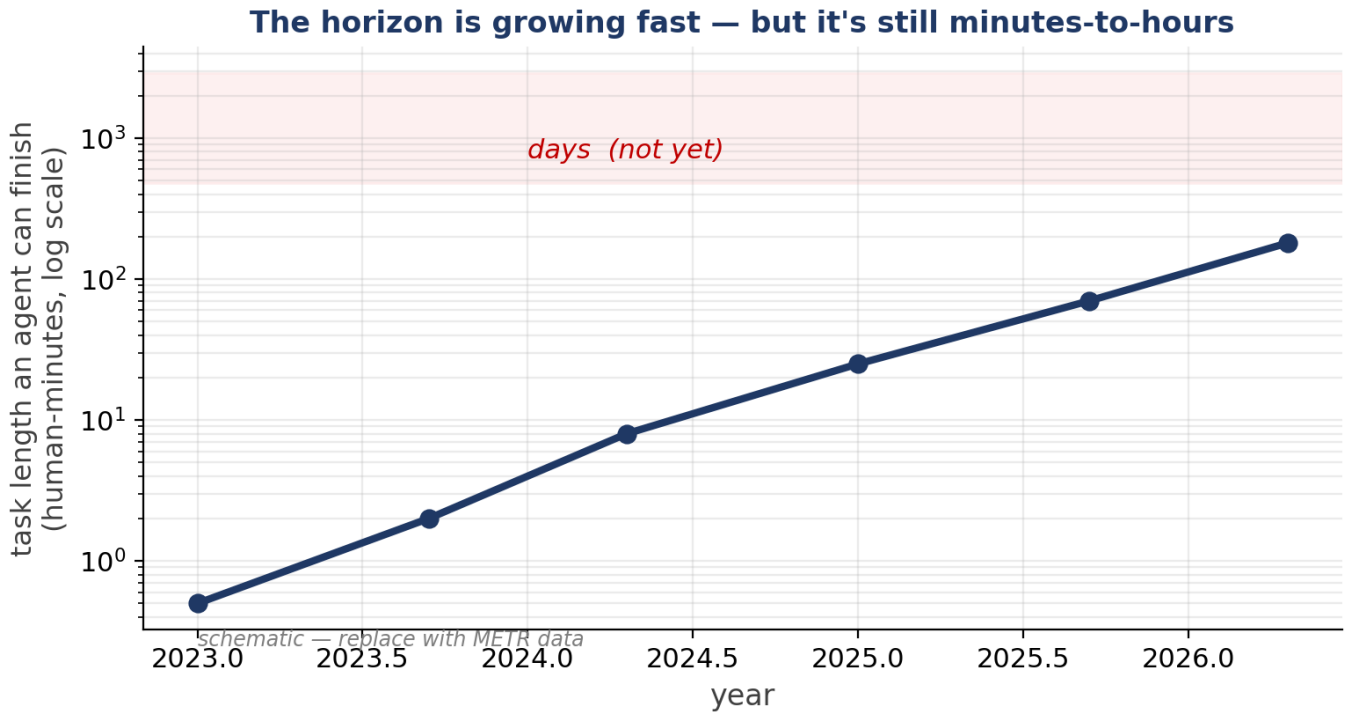
(c)

redoes an already-solved subtask

4. Reality Check

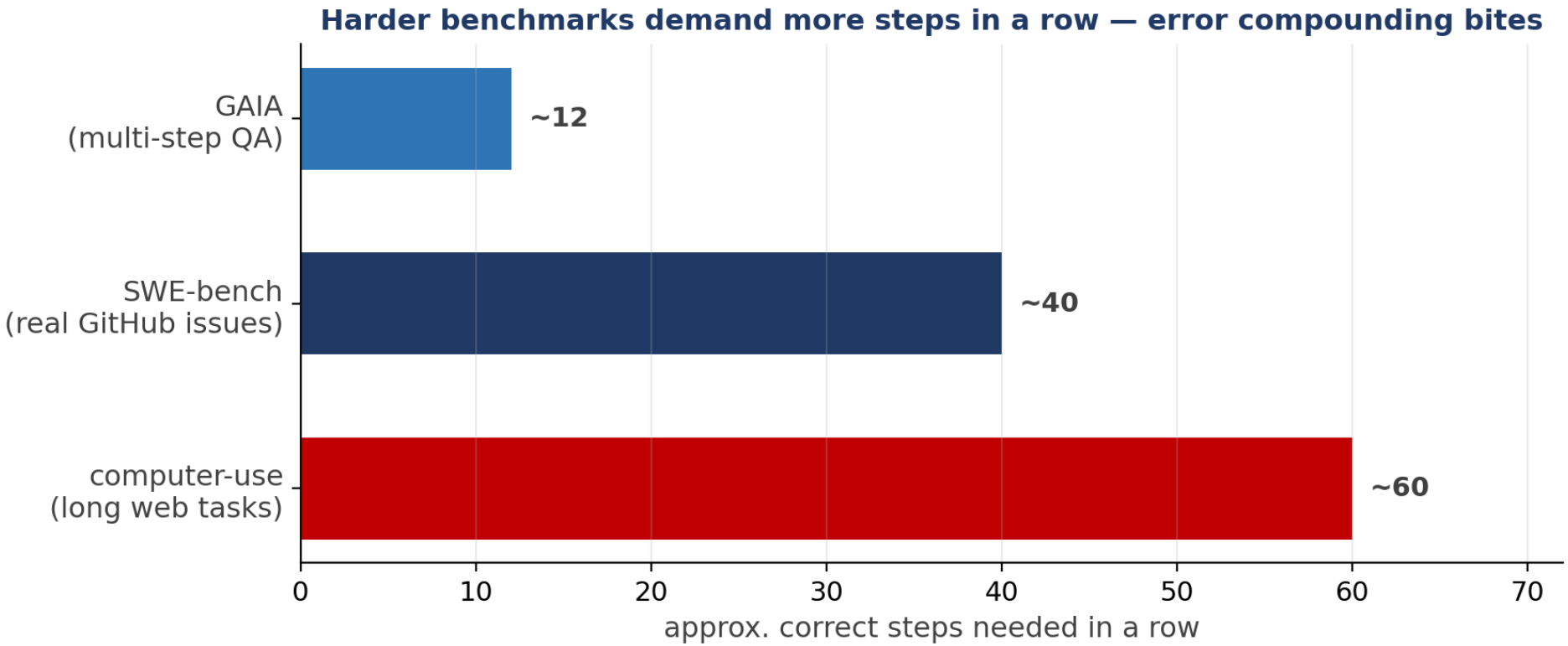
How Long a Horizon, Really?

- **Honest take:** even with all this scaffolding, long-horizon reliability is the field's #1 unsolved problem in 2026.
- **The "time-horizon" lens:** how long a task (in human-minutes) can an agent finish end-to-end?



How We Measure It

- **Long-horizon benchmarks all demand many correct steps in a row:**



More steps in a row → error compounding bites harder (recall slide 7).

What's Still Broken

- Doom loops — repeating the same failing action.
- Silent drift — slowly losing the goal without noticing.
- Confident wrong answers — no internal signal it's off-track.
- Wrong retrieval — memory returns the irrelevant episode.

Summary

- Long tasks fail even with reliable steps — error compounding (p^N).
- Two failure modes: reasoning drift and memory overflow.
- Reasoning fixes: decompose, reflect, search, verify.
- Memory fixes: working → summarized → long-term (RAG over your own history).
- Long-horizon reliability is still the frontier — judge agents by it, not by demos.

Frontier — What's Next

- Agents that learn durable skills from their own traces.
- Memory that truly persists across sessions, not just within one.
- Self-improving agents that get more reliable with use.

The open race is reliability — not the next capability demo.

Next Lecture

L36 made one agent that can sustain a long task. L37 asks: what if one agent isn't enough?

Two questions for next lecture:

- Can several **specialized agents** outperform one generalist?
- How do **humans stay in the loop** when agents run for hours?

L37: Autonomous Systems — Multi-Agent Systems & Human-AI Collaboration