TELOS

Scaling autonomy without sacrificing *quality.*

Effectively leveraging long-horizon coding agents.

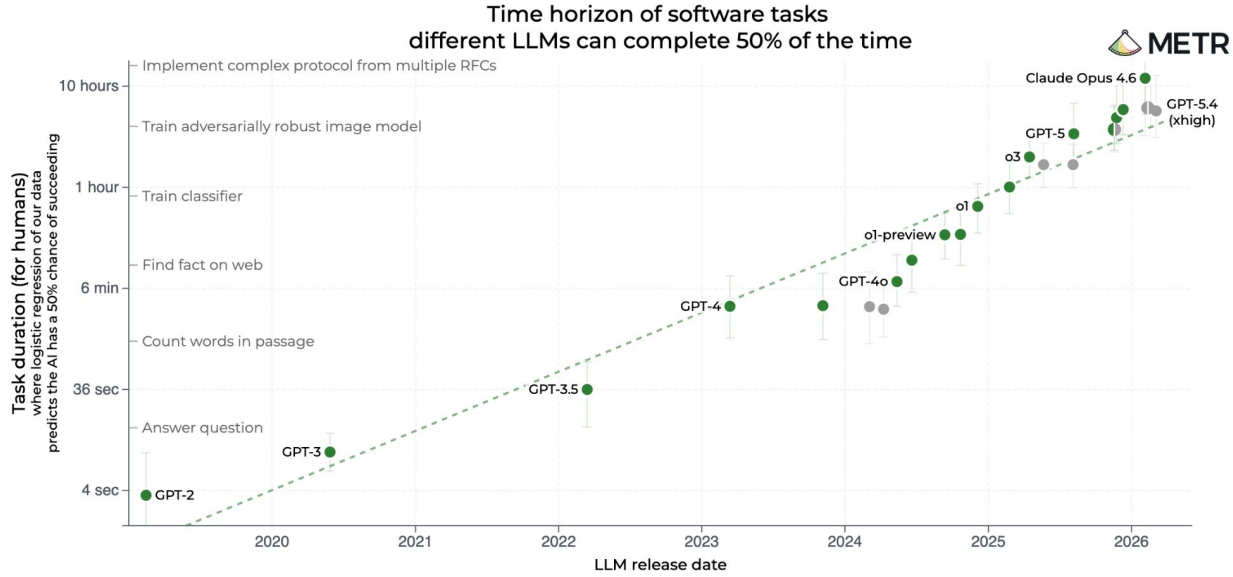
Rohan Gupta

TELOS



An AI-generated image of 'AI slop'. Shutterstock AI Generator

2025: The rise of AI assisted coding, from "vibe coding" to "agentic engineering"



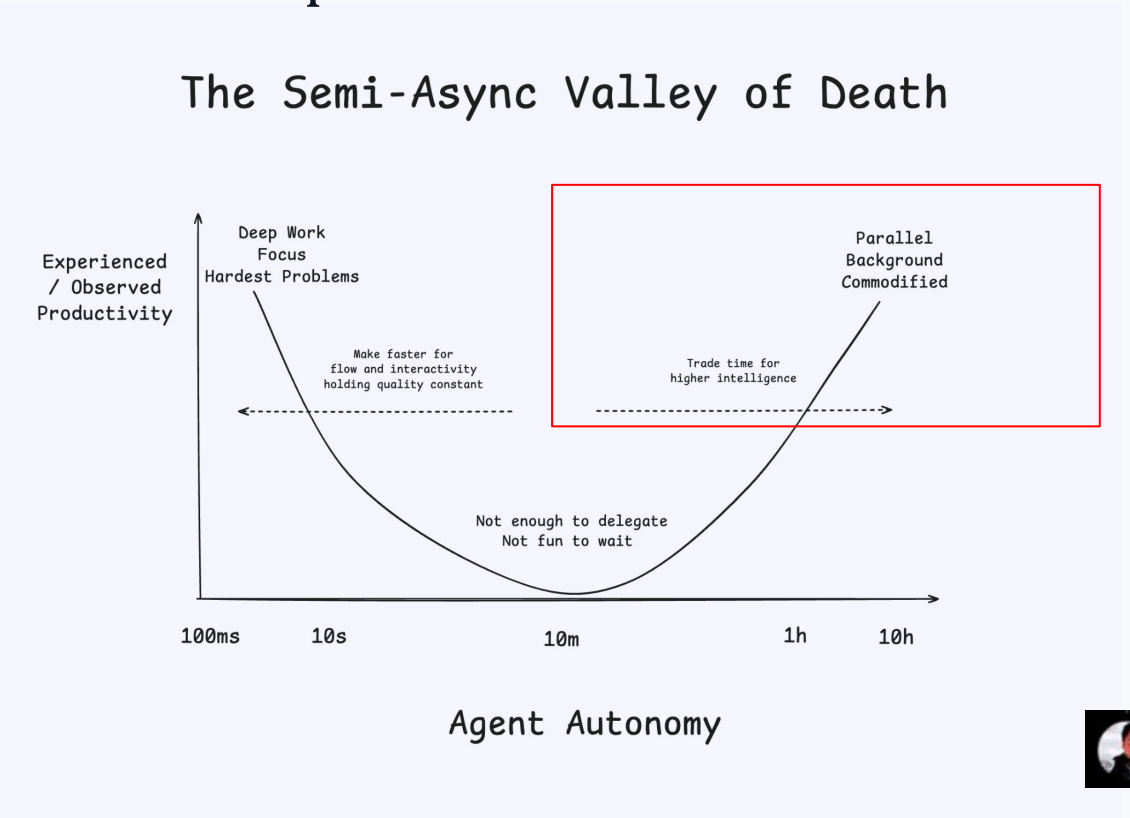
TELOS

2026: ?



TELOS

Not a continuous spectrum





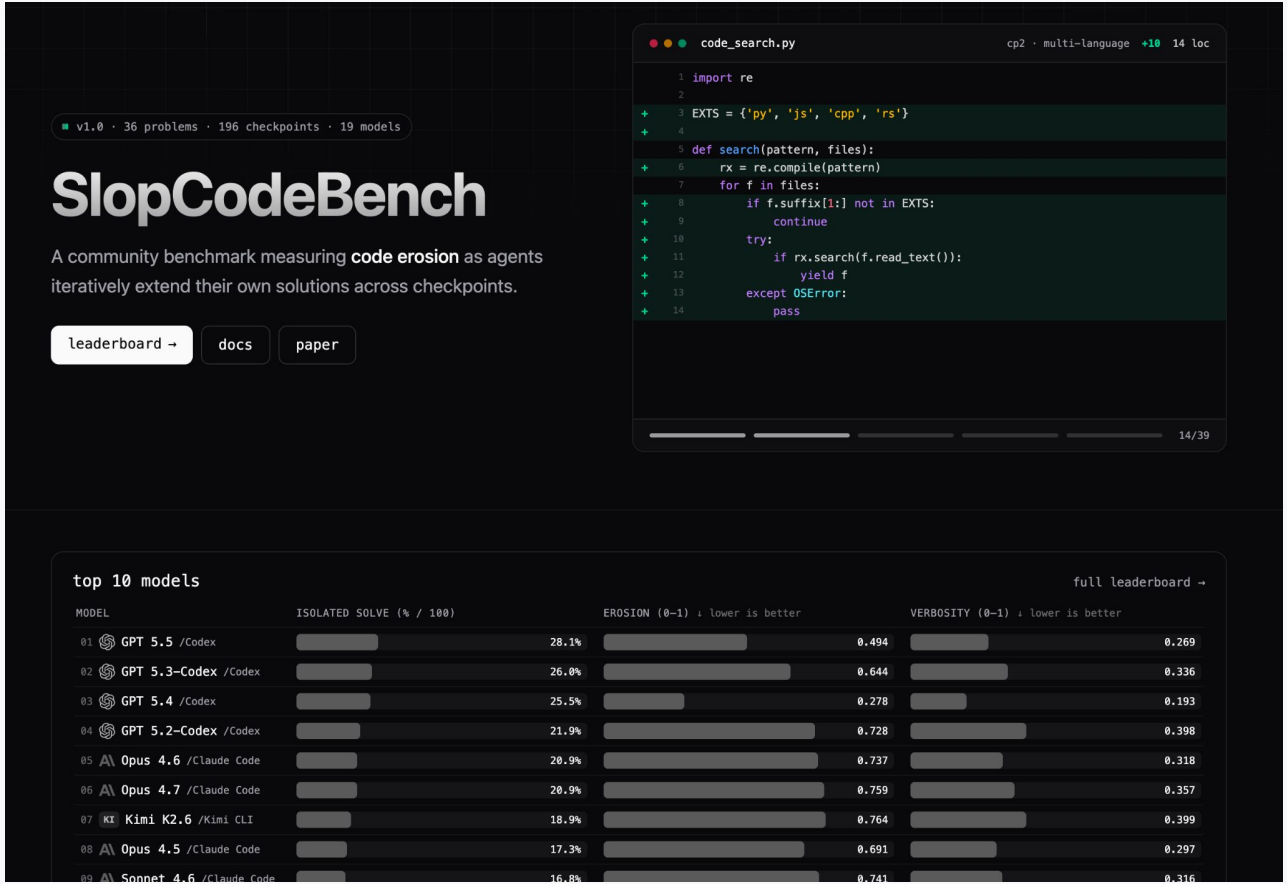
Can you *really* trust your agents to run in the background?

Context rot	Coherence degrades as the window fills.
Sycophancy	Asked to rate its own work, the agent confidently approves it. "You're absolutely right!" is the most dangerous string in agentic coding.
Skill drift	CLAUDE.md, AGENTS.md, your house styles, read but ignored.
Structural erosion	Tests pass. The codebase rots with complexity through incremental iteration.

nico 5 months ago | parent | prev | next [–]

> Claude basically disregards your instructions (CLAUDE.md) entirely

A friend of mine tells Claude to always address him as **"Mr Tinkleberry"**, he says he can tell when Claude is not paying attention to the instructions on CLAUDE.md when Claude stops calling him **"Mr Tinkleberry"** consistently



Turns out *passing tests is not the same as writing maintainable software!*

TELOS

Home > slop-code-v1 > database-migration

Checkpoints

1. checkpoint 1
2. checkpoint 2
3. checkpoint 3
4. checkpoint 4
5. checkpoint 5

Language

Python

Problem Info

Author: Albert Ge

Difficulty: Medium

Category: databases

cli-tools database sqlite schema-migration data-transformation

checkpoint 1

Entry: migration_tool.py

Database Migration Tool - Checkpoint 1: Basic Schema Migrations

Build a CLI tool that applies database schema migrations to SQLite databases. This checkpoint focuses on basic schema operations: creating tables, adding columns, and modifying table structures.

Task

Implement a CLI tool that reads migration specifications from JSON files and applies them to SQLite databases.

Your program should accept:

- Path to a migration file (JSON format)
- Path to a SQLite database file (will be created if it doesn't exist)

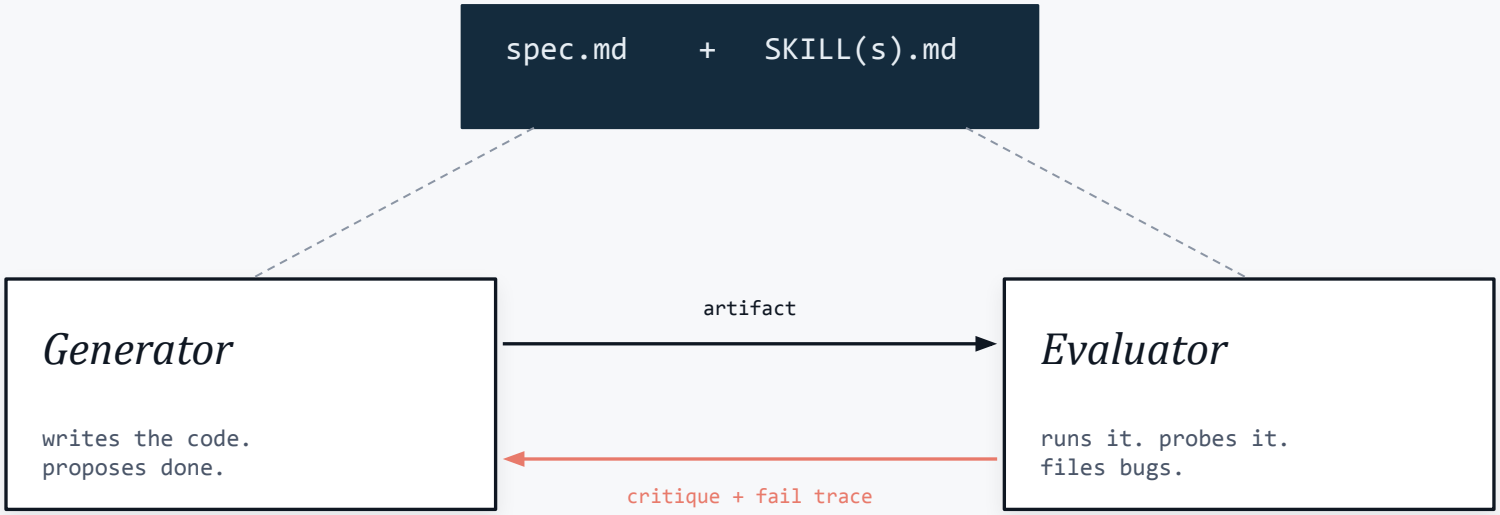
```
python migration_tool.py migrate migration.json database.db
```

Migration File Format

The migration file is a JSON object with the following structure:

```
{
  "version": 1,
  "description": "Initial schema setup",
  "operations": [
    {
      "type": "create_table",
      "table": "users",
      "columns": [
        {"name": "id", "type": "INTEGER", "primary_key": true, "auto_increment": true},
        {"name": "email", "type": "TEXT", "not_null": true, "unique": true},
        {"name": "name", "type": "TEXT", "not_null": true},
        {"name": "created_at", "type": "TIMESTAMP", "default": "CURRENT_TIMESTAMP"}
```

Proposed solution: pair your agents!



TELOS

Demo

*https://www.anthropic.com/engineering/harness-design-long-running-apps *https://openai.com/index/prover-verifier-games-improve-legibility/

Telos: a spec-driven agent runtime

Run the generator evaluator loop on specs on your machine, with the coding agent you already pay for.

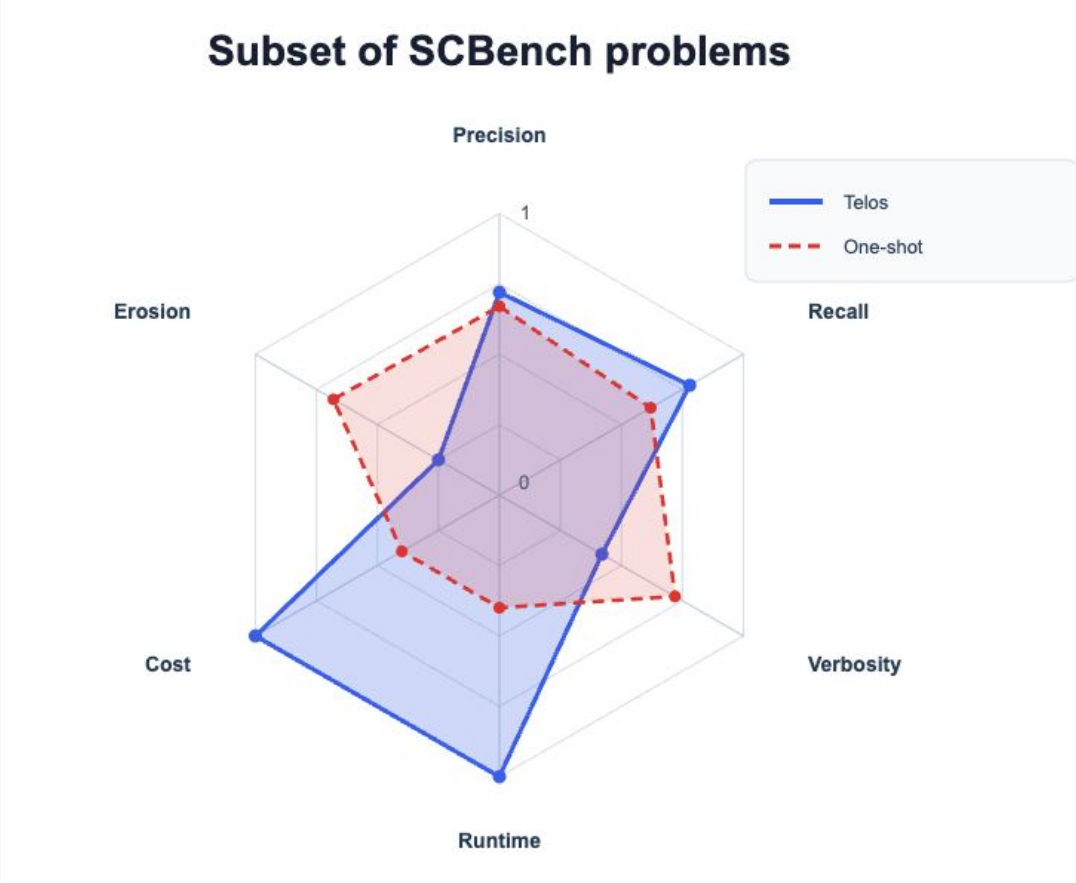
```
$ telos run ./spec.md

# reads spec.md
# generator → verifier → loop until done in the background

# apache-2 · github.com/telos-org/telos
```

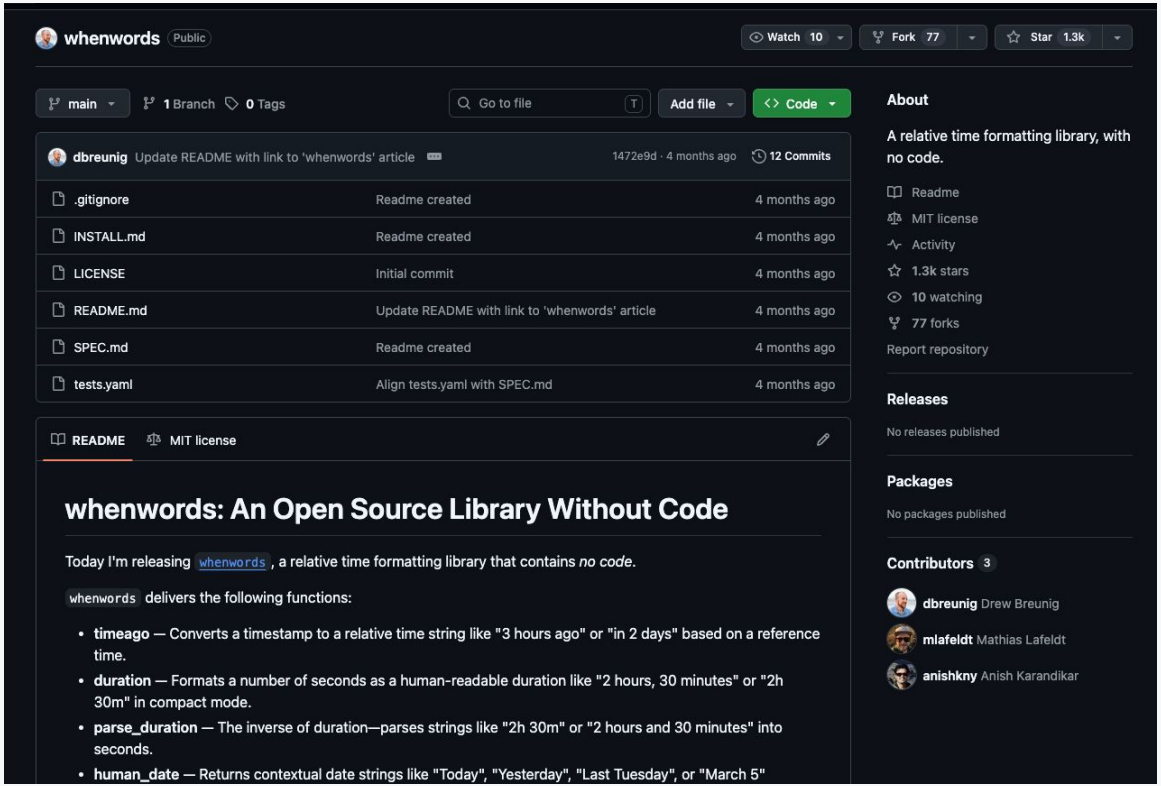
~/your-repo

● ready


Subset of SCBench problems
Precision
Recall
Verbosity
Runtime
Cost
Erosion
1
0
Telos
One-shot

Plan high-level goals. *Then go for a walk.*

Coverage	Goal: "every public function has a meaningful test, line coverage >= 90%" Run. Come back to a PR.
Migrations	Spec the new schema, the new API, the deprecation.
Performance	Spec the regression budget. The loop hill-climbs against it. Wake up to a faster service.
End-to-end generation	Spec to code generation. AGENTS.md /skills become part of the spec. Enforce process boundaries.

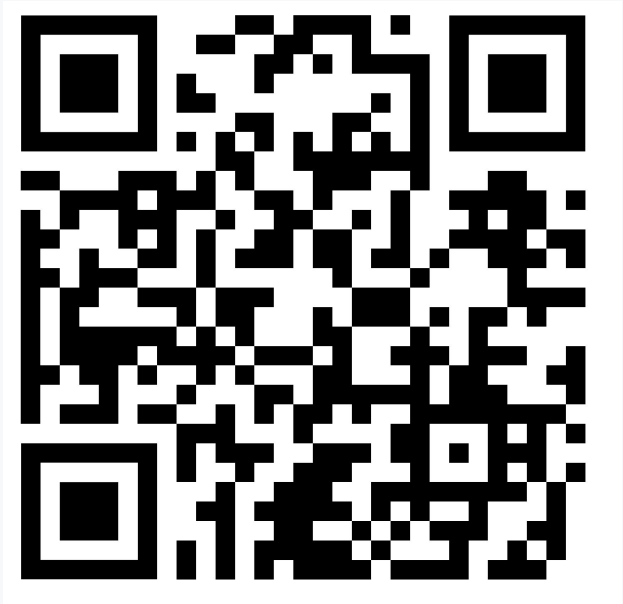


https://www.dbreunig.com/2026/01/08/a-software-library-with-no-code.html

TELOS

Thank you!

LinkedIn: in/rohan-gupta2
X: @rohangupta_



https://github.com/telos-org/telos