

Evals for Coding Agents

Evaluating AGENTS.md and CLAUDE.md with Harbor

Jeff Linwood · Zebric & RobotBridge

Austin AI MUG, October 5, 2026

<https://www.jefflinwood.com/>

@jefflinwood

CONTEXT

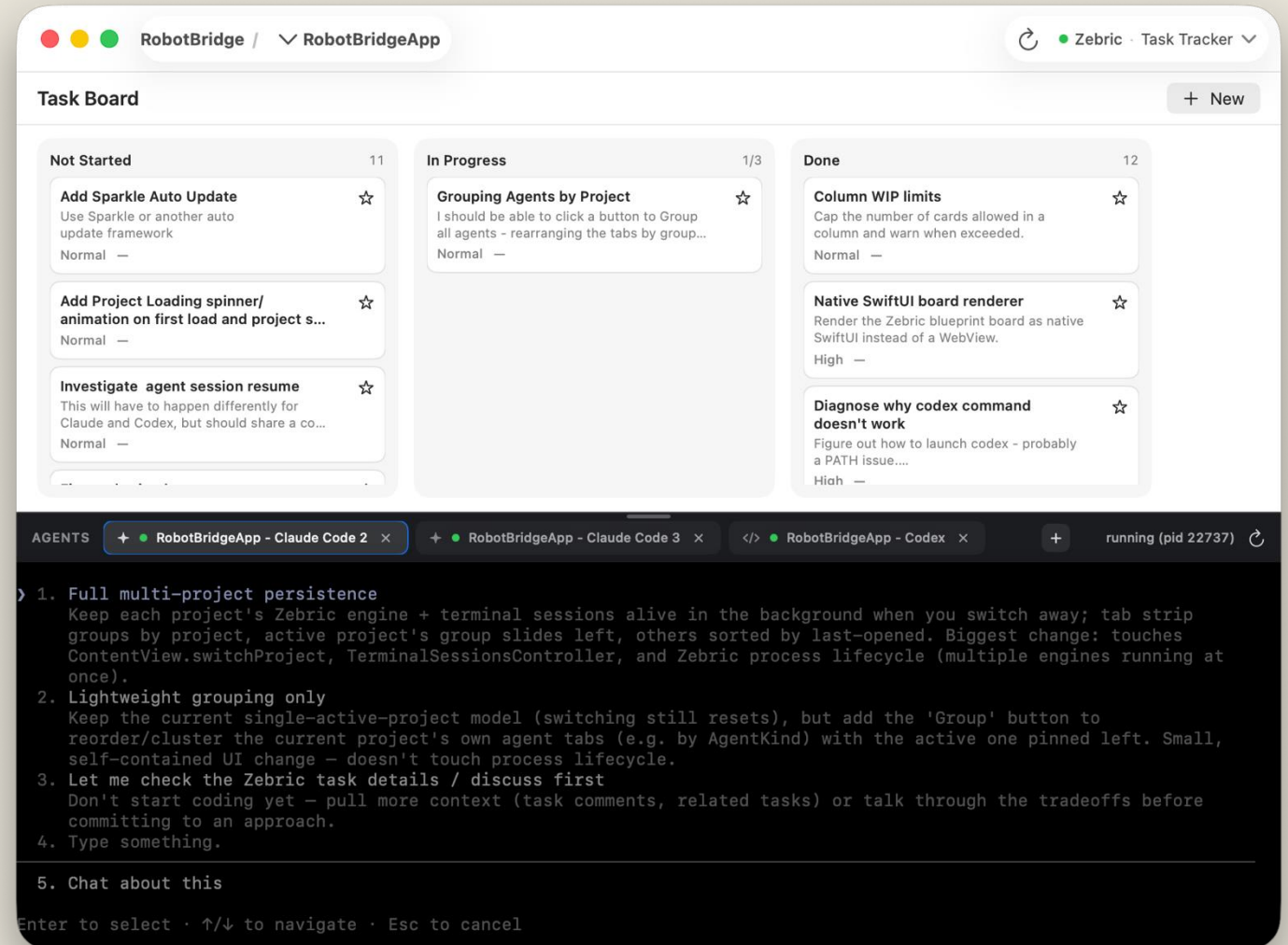
RobotBridge

My open source macOS agent workspace built on a **Zebric blueprint**, so humans and agents work together from the same tasks.

Task board for people

Claude Code and Codex in terminal tabs

Zebric MCP connects the two



The agent never called the MCP tools

> Work on the "Update MCP description on README" task

- Read README.md
- Search for "MCP"
- Edit README.md

Zebric MCP calls: 0

The task lived on the RobotBridge board, backed by Zebric.

Its details were one MCP call away — and Claude Code and Codex both had the tools.

They worked from the title alone, with no idea where the context was.

The fix is a few lines in *AGENTS.md*.

What's the right wording? How do we test it?

An eval is a repeatable test for behavior

Task

A realistic prompt and a known starting state

Agent

The harness and model pair under test

Verifier

Code that checks what happened and scores it

Trials

Many runs, because agents are nondeterministic

Harbor runs real agents against tasks in sandboxes

<https://www.harborframework.com/>

Runs the actual CLIs: Claude Code, Codex CLI and more

Docker locally, or can host on cloud-provided instances (such as Daytona)

I used API credits, but you may be able to use subscriptions

```
$ uv tool install harbor
```

```
$ harbor run -p ./tasks \  
  --agent claude-code \  
  --model anthropic/claude-sonnet-5 \  
  --n-concurrent 4
```

A task is a folder

readme-mcp-task/

└─ instruction.md

└─ task.toml

└─ environment/

| └─ Dockerfile

└─ tests/

└─ test.sh

instruction.md

The prompt the agent gets, exactly as a user would type it

environment/

The repo, a Zebric runtime, MCP config — and the AGENTS.md variant

tests/

The verifier: hidden pytest plus Zebric's audit log and DB

What others have measured

Agents follow context files

Naming uv in AGENTS.md took it from ~0 to 1.6 calls per task. Advice: keep files to minimal requirements.

Gloaguen et al., ETH Zurich, 2026

More instructions, less followed

Accuracy falls as instruction count grows. Smaller models fall fastest.

IFScale, Jaroslawicz et al., 2025

Can agents use MCP tools?

Benchmarks test correct tool use when the task calls for it.

MCP-Bench, MCP-AgentBench, 2025

This experiment

Do agents use an available workflow tool when nobody tells them to?

Do instructions in AGENTS.md/CLAUDE.md get agents to use the Zebric tracker?

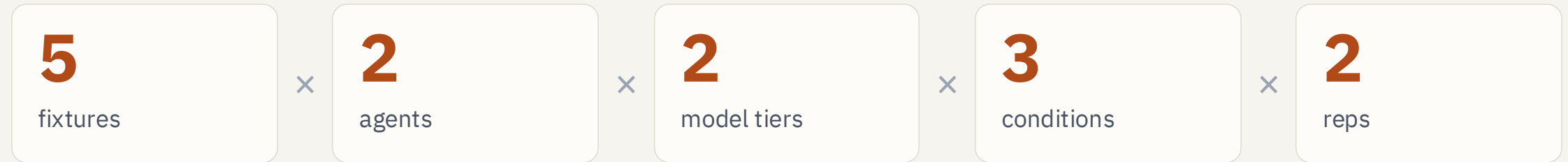
Question 1

Did it write the code?

Question 2

Did it use the MCP tools?

120 runs - turns out this was enough



Tier	Claude Code	Codex
small	claude-haiku-4-5	gpt-5.6-luna
medium	claude-sonnet-5	gpt-5.6-terra

Conditions

Control
Short instruction
Product instruction

Scored from evidence

Signal	Source of truth
MCP used / task discovered	Zebric application's audit log
Task claimed / completed	Status timeline + final Zebric DB
Coding success	Hidden pytest run by the Harbor verifier
Invalid run	Preflight failure or harness error

Primary outcome

Did the seeded task end in **done**?

No LLM-as-judge.
No self-reporting.
No Jev :)

Tasks completed in the tracker

Agent	Tier	Control	Short instruction	Product instruction
Claude Code	small	0/10	0/10	2/10
Claude Code	medium	0/10	10/10	3/10
Codex	small	0/10	2/10	2/10
Codex	medium	0/10	8/10	0/10

All 120 runs passed the coding tests.

05 · RESULTS

120/120

runs passed the coding tests

0/40

control runs completed the task in the
tracker

Control - no instructions in AGENTS.md/CLAUDE.md

Short instruction

This project has a Zebri MCP task board (server `zebric`: `list\_tasks`, `get\_task`, `set\_task\_status`, `create\_task`, `update\_task`, `delete\_task`). Before starting work on a task or or anything board-related, call `list\_tasks` first. Set the task to to `in\_progress` when you start and `done` when you finish.

20/40

Medium tier: 18/20

RobotBridge 0.2.0 instruction (excerpted)

This project registers a Zebri MCP server (tool names are prefixed prefixed `zebric\_`). It exposes:

- **task_tracker** — Inspect and manage tasks in the Task Tracker application.
 - `zebric\_task\_tracker\_list\_tasks` — List tasks, optionally filtered by status or priority.
- ... <more tools>

Board "Task Board" tracks Task grouped by `status`: Not Started (`not\_started`), In Progress (`in\_progress`, WIP ≤3), Done (`done`)

7/40

Medium tier: 3/20

Takeaways from the Experiment

Making tools available wasn't enough

Control: 0/40 tools called

Shorter instructions performed well

Short instruction 20/40, product instruction 7/40

Small models worse than medium

Small tier with either instruction: 6/40

Had to measure Zebric calls

120/120 passed tests — a code-only eval would show nothing wrong

These numbers are from a small ~\$20 experiment, not a benchmark-style ranking

What you could eval with agents and Harbor

Instruction files

Does a change to AGENTS.md, CLAUDE.md or a skill change behavior?

MCP tool design

Which names and descriptions actually get picked

Harness and model choice

Same tasks through Claude Code and Codex before you standardize

Regressions

Rerun the suite on every prompt, tool or model upgrade

Your own benchmark

Turn real tickets from your repos into tasks

Training data

Harbor can also be used for reinforcement learning

THANK YOU

Links

RobotBridge

robotbridge.com

github.com/RobotBridge/RobotBridgeApp

Zebric

zebric.dev

github.com/ZapCircleHQ/zebric

Harbor

github.com/harbor-framework/harbor
