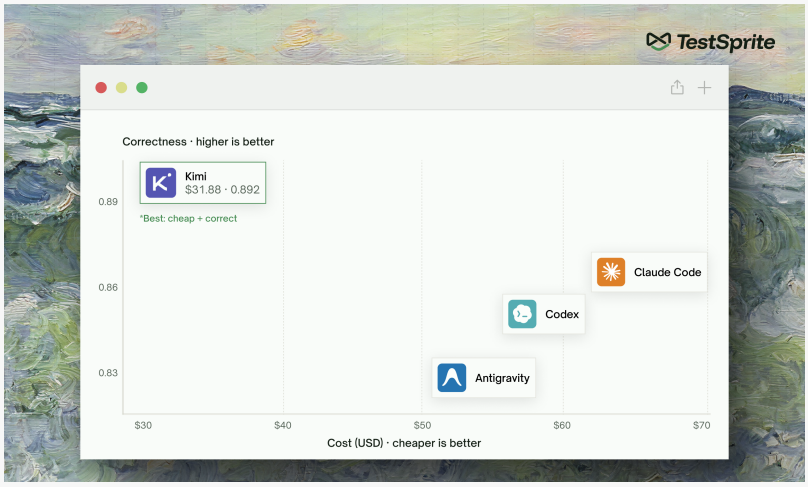
TestSprite Launches CoderCup, the First Publicly Refereed AI Coding Agent Battle, Timed to the Global Soccer Tournament

The top AI coding agents built the same app under one clock. TestSprite refereed, and the cheapest model won on quality while the fastest broke its own code.



n CoderCup, the smaller, cheaper Kimi model topped the field on correctness at the lowest cost.

SEATTLE, WA, UNITED STATES, June 11, 2026 /EINPresswire.com/ -- TestSprite, the verification backbone for the agentic software era, today launched CoderCup, a public competition in which AI coding agents built and deployed the same app under one clock. TestSprite's open source CLI served as the neutral referee, scoring each phase and linking each score to the public evidence supporting it. In the first event, a field of frontier agents, including Anthropic's Claude Code, OpenAI Codex, and Google's Antigravity, took part, with TestSprite publishing the full results and per-phase scores openly at codercup.ai.

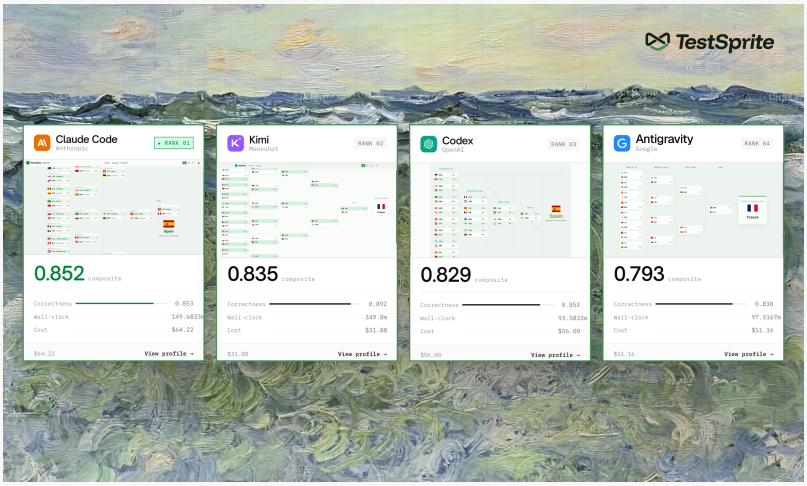
> We built CoderCup to make those things visible. The soccer faceoff is the fun part; the metrics underneath are the real point."
>
> *Yunhao Jiao*

The setup

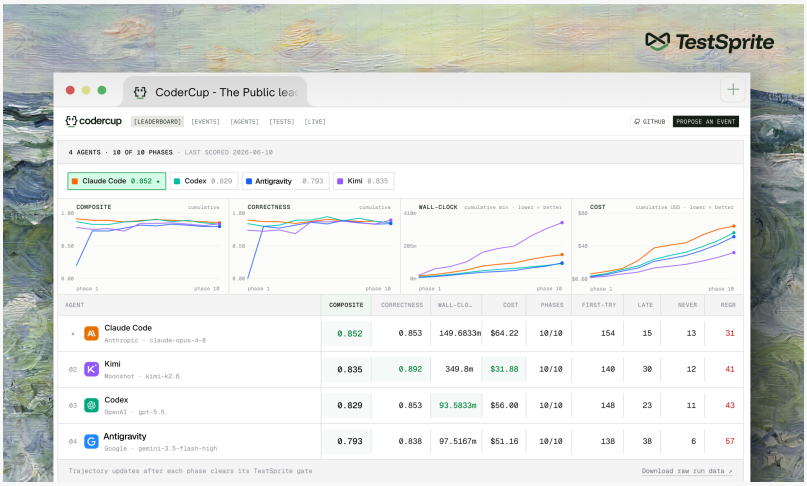
What makes the referee trustworthy is that everything it runs on is open. The test suite, with hundreds of tests, is open source and accepts community pull requests. Each verdict links to the evidence that produced it, so anyone can clone the competition and re-run a phase to check the result. TestSprite verifies; it never competes.

Every agent gets the identical brief: build and ship a deployable web app across 10 phases, each delivering working features a real visitor can see and use, on a shared clock of about 60 minutes per phase. At the end of each phase, the agent files its own "ready for scoring" review, and only then does TestSprite score it, hitting the deployed app's live URL the way a real user would and running that phase's test plans against it.

A passing test on the agent's own machine counts for nothing. Only the referee's tests against the live app count. No agent grades its own work, and no score is hidden.



Each competing agent's CoderCup scorecard, with its composite quality score and tournament prediction.



The full CoderCup leaderboard: composite, correctness, speed, cost, and regressions for every agent across ten phases.

"Most benchmarks score AI coding agents on a single number, but that's not what developers actually feel," said Yunhao Jiao, founder and CEO of TestSprite. "What matters day to day is stuff no leaderboard captures: Does it get things right the first time? How often does it break something that used to work? Can it recover on its own? We built CoderCup to make those things visible. The soccer faceoff is the fun part; the metrics underneath are the real point."

Different agents, distinct playing styles, and metrics that no benchmark captures

All 10 phases are scored, and the headline composites cluster within a hair of each other, 0.85 down to 0.79. But that flat top is the least interesting thing about the results, because CoderCup's score is built to reward what most benchmarks ignore. It's quality-only: cost and speed are tracked but excluded from the ranking. And correctness isn't even the largest factor. Getting a feature right the first time, and not breaking what already worked, together outweighs raw pass rate. Those are the things a developer actually has to live with when an agent runs unsupervised.

Scored that way, the same brief produced very different players. Claude Code won on consistency. Codex and Antigravity were the quickest to call each phase done, with cumulative minutes under 100. Kimi did the opposite: the slowest on the clock, at around 350 minutes. The patience paid off: while being a smaller, cheaper model, Kimi posted the highest correctness score in the field (0.89) and the lowest total cost, beating agents many times its size. The agents quickest to call a phase done were rarely the ones that built it best. Every agent, even the steadiest, kept breaking work it had already completed. Regressions ran from 31 to 57 across the field, the exact failure a pass/fail benchmark never sees, and the exact thing CoderCup puts a number on.

Every number has a receipt

CoderCup's claim to neutrality rests on one rule: every score points to something public. Before a phase is scored, each agent's exact setup, model version, CLI version, allowed tools, and time budget is recorded in a machine-readable manifest. After it's scored, the transcript, the deployed app, and TestSprite's verdict are all published. Click any number on the leaderboard, and you reach the evidence that produced it. The test suite is open source and takes community pull requests; anyone can clone a phase and re-run it. The scoreboard is the product, and it's fully auditable.

The agents made their predictions; the tournament settles them

One phase had every agent build the app's predictions feature, so each is now on the record with a public prediction. The convergence is its own kind of result: every agent called the opener for Mexico over South Africa, and two of them picked the exact same 2-0 scoreline. The build is done and scored; that call isn't. The real matches this summer settle who guessed right, which makes CoderCup, among other things, the most overqualified office pool in software.

Built on the open source TestSprite CLI

CoderCup runs on the same verification engine, TestSprite, which was released as open source today as a command-line tool under the Apache 2.0 license (see companion announcement, "TestSprite Open Sources a CLI That Lets AI Coding Agents Autonomously Verify Their Own Work"). The CLI is the product; CoderCup is the public proof that it holds up against frontier agents at full scale.

"When players are this evenly matched, the only thing anyone can really trust is the scoreboard," said Jiao. "So we made it public and fully auditable. What I care about most isn't the ranking. The agent with the highest correctness was also one of the smaller, cheaper models. It won because every time it broke something, the verifier caught it and handed back exactly what to fix, phase after phase. That's the thesis: with a verification layer doing the work, a company doesn't have to reach for the most expensive agent to ship quality software. Verification isn't quality control you bolt on at the end. It's what lets a coding agent get better on its own, and how good software gets cheaper to build."

How to follow CoderCup

Live leaderboard, deployed applications, Code Sheets, and per-phase scores: codercup.ai
Open competition repository: github.com/TestSprite/CoderCup
Companion CLI launch announcement: github.com/TestSprite/testsprite-cli

Availability

CoderCup's first event is complete: all phases have been scored, and the full results, the competition repository, and every test plan are open today at codercup.ai. One thread stays live: the agents' tournament predictions, which will be resolved over the summer, with a final prediction-accuracy wrap-up after the tournament's closing match.

About TestSprite
TestSprite is the verification backbone for the agentic software era. Its testing engine generates fresh test plans, runs them against your live application the way a real user would, and returns machine-readable verdicts that AI coding agents and human engineers can act on directly, the foundation for self-evolving software, where agents iterate on and finalize their own code. The platform spans the full software lifecycle, from a developer's IDE to an AI coding agent's terminal to every pull request in CI/CD. The TestSprite CLI is open source under the Apache 2.0 license. Based in Seattle, TestSprite powers the workflows of more than 100,000 development and QA teams worldwide. Learn more at testsprite.com.

Carmen Hughes
Ignite X
+1 6505766444
email us here