

The Opus 5.5 Stress Test & The 401 Disaster

Goal: Evaluate Opus 5.5 reasoning efforts (Medium vs. High).

Scope: 22 real commits replayed across 11 repositories (TypeScript, Rust, Python, Node).

Method: Fresh git worktrees, hidden tests, identical prompts.



Halfway through testing, a revoked local token killed 39 agents in 12 seconds.

Workflow subagents don't retry on 401.

Only 44 complete pairs survived to form our final dataset!

The Illusion of 100%: Why Tests Aren't Enough

Hidden Tests Passed: 44/44
(Both Medium & High)

- ☁ **The Flaw:** Medium relies entirely on tests. It scaled bar heights against uncompressed series while drawing compressed bars.
- ☁ **The Result:** The 90-day and all-time charts overflowed past 100%.
- ☁ **The Lesson:** Rendering wasn't tested, so both runs went green. High thinks one step further to catch what the tests miss.



The Blind Verdict



Every pair judged blindly with A/B swapped to cancel position bias.

The Economics of Survival

Time: 1.33x slower (234s vs 140s median)

Tokens: 1.35x more output tokens

Cost: ~1.29x more expensive



Takeaway: High buys +3 Index points for ~1.3x the cost. It sits perfectly on the intelligence-vs-cost Pareto frontier before returns shrink drastically.

The Config Trap & Final Recommendations



Beware ultracode!

If **'ultracode': true** is set, every session is forced to xhigh. Your **effortLevel** setting is silently ignored. Turn it off to make **High** your default.

No Human Review/ Hard Problems

-> High / xHigh

Catches the silent bugs that tests miss when nobody reads the diff.

Pair Programming (Reading Diffs)

-> Medium

1/3rd faster, stays in scope. You'll easily catch the bugs it misses.

Tight on Quota / Pro Tier

-> Medium

The 35% token gap adds up fast. Pass --effort high only for hard tasks.

Bulk Mechanical Work

-> Low / Medium

Thinking harder doesn't help simple chores like greps or renames.