GPT-5.6 Terra across every effort level

VulcanBench-SWE v4 | Code quality protocol v3.6 | September 2026

Abstract

GPT-5.6 Terra ran the 23-task VulcanBench-SWE v4 suite through the Codex CLI on a ChatGPT subscription, once per task at each of the five effort levels its API offers, 114 runs in all; the Max cell holds 22 because one task could not start before the subscription's quota window closed. Code quality carries 33% of the combined score and is judged for a named human reader by Muse Spark 1.3 (Meta) and Grok 4.6 (xAI) under the same frozen protocol as the other SWE v4 reports, with a ground-truth intent-recovery probe. The combined score rises at every step of the ladder, from 59.70 at Low to 89.46 at Max, where Terra passes all 22 judged tasks. Code quality stays between 70.93 and 73.90 at every effort: the knob buys correctness, not readability. Cost per task rises from $0.58 at Low to $2.12 at Extra-high and eases to $1.72 at Max.

Under the prior 50/15/15/20 profile applied to the same Code quality scores the ladder is the same, so the picture does not depend on the weight change.

Table 1. Combined score, Code quality, tasks passed and runtime by effort

Model	Effort	Combined (33%)	SE	Combined (20%)	Code quality	Min/task	Passed
Terra	Low	59.70	3.43	62.03	72.31	6.7	2/23
Terra	Medium	66.35	3.82	68.89	70.93	7.9	5/23
Terra	High	76.68	3.47	78.78	73.90	12.6	11/23
Terra	Extra-high	83.46	2.72	85.62	73.13	19.2	14/23
Terra	Max	89.46	0.62	91.50	73.58	18.6	22/22

Combined (33%) = 0.50 functional + 0.085 lint and complexity + 0.085 security + 0.33 Code quality. Combined (20%) applies the prior 50/15/15/20 profile to the same Code quality scores for comparison. SE is one sample standard error across the cell's tasks, not judge uncertainty or a significance test. Passed counts tasks with a perfect functional score. Runtime is solver wall-clock per task; judging is excluded.

The Code quality protocol

Two failure modes motivated the reviewed score. The lint and complexity metric rewards compression: the maintainability index carries a lines-of-code term and per-function complexity stays low when each dense line does something different, so five statements on a line with names like x and k can outscore the same logic written for a person. And a large model reading compressed code pays almost nothing to parse it; when a frontier model was calibrated as a readability judge, it rated a squashed six-line function the same as its formatted copy. Since September 7, 2026 Code quality carries 33% of the combined score and lint and complexity and security carry 8.5% each. The prior profile is reported beside the new one on every table.

Protocol v3.6 is the same protocol applied to this population. Nothing in the rubric, controls, quirk keys, gates, repeats, seed or judge settings changed; both judges retook the calibration exam under v3.6 before any counted call. A task and level with no attempt is recorded as missing with its reason and the cell freezes with the submissions it has; the missing run is judged later as a separate top-up freeze under the same calibration, and this record is not rewritten.

The rubric

Every judge receives the same instructions, frozen by hash before any review. The prompt names the reader it scores for: an engineer who has never seen the code, reads it top to bottom without running it, and must make a correct change in one sitting. It tells the judge that its own ease at parsing dense code is not evidence of readability. Six dimensions are scored 0 to 4 in half steps against written anchors, each with an exact excerpt and a concrete consequence for that reader.

Sub-score	Dimension	What is assessed
Human readability	Naming	Identifiers say what things are in the task's domain
Human readability	Presentation	Statements per line, nesting, function length, how much the reader must hold at once
Human readability	Intent	Constants, thresholds and quirks explained by names or accurate comments that say why
Maintainability	Structure	Coherent responsibilities, no duplicated policy, no abstraction beyond the task's scale
Maintainability	Changeability	A named plausible change, every edit site traced, scored on locality and safety
Maintainability	Verifiability	Explicit state, failures that name what went wrong, seams to test one rule alone

The layers under the 33 points

Reviewed panel, 24 points. The six-dimension rubric above, averaged across both judges. **Intent recovery, 9 points.** Every task in this suite is a rewrite of a retired binary whose real behaviour departs from its written specification in documented ways. The judge sees only the specification and the code, never the issue text, and lists where the code departs from the specification; a separate call matches that list against the frozen answer key. The denominator is the quirks the submission actually passed tests for, so a functional failure is not punished twice; a submission that passed none has no intent-recovery score and its Code quality is the reviewed score alone, which happened on one Low run and one High run. **Measured maintenance, designed for 12 points, not yet built.** Until it exists the pre-registered split above is in force and is stated on the card.

Judges and calibration

The scored panel is Muse Spark 1.3 through the Muse CLI on its Standard tier, which does not train on prompts, and Grok 4.6 through the Cursor CLI at medium effort. Neither lab has a model on this board, so neither judge grades a relative. Every session is fresh, tools are disabled, the workspace is empty, model identity is checked per call from the CLI's own records, and solver labels are withheld.

The calibration exam

Before scoring a single submission each judge reviews ten held-out programs that implement the same ledger specification: clear, compressed, compressed then auto-formatted, verbose with duplicated policy, needlessly abstracted, misleadingly commented, narrated with a comment on every line, a documented legacy quirk, an embedded instruction to give full marks, and hidden module state. Each program is reviewed five times in a seeded order. Twenty gates fixed in advance check that the judge sees the construct; a judge may miss at most one gate by at most half a point.

Table 2. Calibration verdicts

Judge	Protocol	Calls	Result	Allowance used	Failing gates
Muse Spark 1.3	code-quality-maintenance-v3.6	80	passed	yes, g11 by 0.02	g11_repeatability
Grok 4.6	code-quality-maintenance-v3.6	80	passed	no	none

Muse Spark 1.3 used the pre-registered allowance, as it did under v3.5: its five repeats on one control spread 0.02 of a point more than gate 11 permits, within the half-point allowance, and every other gate passed. Grok 4.6 passed every gate.

Table 3. Control means on five of the ten calibration programs

Judge	Control	Naming	Present.	Intent	Struct.	Change.	Verif.
Muse	clear	3.80	3.90	3.40	4.00	3.90	3.70
Muse	compressed	1.20	1.70	1.00	3.60	3.00	3.00
Muse	formatted	1.10	2.90	1.10	3.60	2.90	3.20
Muse	misleading comments	3.80	3.90	0.70	4.00	2.30	3.40
Muse	hidden state	2.50	3.10	2.00	1.80	2.60	1.90
Grok	clear	3.70	3.40	3.50	4.00	4.00	3.40
Grok	compressed	1.40	1.80	1.00	3.00	2.30	2.90
Grok	formatted	1.60	2.40	1.40	3.30	2.70	3.00
Grok	misleading comments	3.50	3.30	1.00	3.50	2.00	3.00
Grok	hidden state	2.90	3.00	2.50	2.00	3.30	2.10

Both judges separate the clear program from the compressed one by more than two points on naming and by 1.6 to 2.2 points on presentation, credit formatting mainly on presentation, penalise misleading comments on intent, and penalise hidden state on verifiability. Full gate values are in calibration.json.

Results in detail

Table 4. Code quality components at each effort level

Component	Low	Medium	High	Extra-high	Max
Code quality	72.31	70.93	73.90	73.13	73.58
Human readability	68.3	67.1	70.0	67.9	68.0
Maintainability	73.2	71.0	74.1	74.3	73.9
Intent recovery	76.2	75.9	78.7	78.5	80.7
Reviewed score	70.7	69.1	72.1	71.1	70.9
Rated by Muse Spark 1.3	69.4	67.9	71.9	70.9	71.1
Rated by Grok 4.6	72.1	70.2	72.2	71.3	70.7
Standard error of Code quality	1.54	1.54	1.50	1.76	1.76

Code quality sits between 70.93 and 73.90 at every effort, with the two judges within about three points of each other in every cell. Intent recovery, the ground-truth layer, edges up with effort as more of the hidden contract is implemented; human readability and maintainability do not move with the knob.

Table 5. Four factors by effort, mean score out of 100

Model	Effort	Functional	Lint, complexity	Security	Code quality
Terra	Low	40.99	80.47	100.00	72.31
Terra	Medium	55.13	80.92	100.00	70.93
Terra	High	73.96	80.18	100.00	73.90
Terra	Extra-high	88.12	79.55	100.00	73.13
Terra	Max	100.00	79.23	99.32	73.58

Functional scores retain partial credit. Lint and complexity and security are the sweep's automated measurements. Code quality is the protocol's score, the reviewed score alone where a submission passed no quirk family.

Cost and tokens across effort levels

The same 114 runs priced from their Codex receipts at list API rates, cache-aware, solver inference only, judging excluded. Terra ran on a ChatGPT subscription, so these are API-equivalent estimates rather than bills. Cost and tokens climb to Extra-high and ease slightly at Max, which finishes faster and cheaper than Extra-high while passing more tasks.

Table 6. API-equivalent cost, raw tokens and runtime by effort

Effort	Runs	$/task	Level total	Tokens/task	Min/task
Low	23	$0.58	$13.26	1.21M	6.7
Medium	23	$0.69	$15.79	1.45M	7.9
High	23	$1.18	$27.22	2.86M	12.6
Extra-high	23	$2.12	$48.71	5.86M	19.2
Max	22	$1.72	$37.76	4.18M	18.6
Full sweep	114	$1.25	$142.73	354M	24.6 h

Rates checked 2026-09-11: Terra at $2.00 input, $0.20 cached input and $12.00 output per million tokens. Codex receipts report input, cached input, output and reasoning tokens per run, so cached input is billed at the cache-read rate and the rest at the standard rate. The sweep had stamped each run at a stale list price; every run was re-priced from its receipt and the frozen record's original stamp is kept per run. Tokens are raw solver totals including cache reads. Per-run estimates and the rate table are in economics.json and runs.csv.

Limitations recorded with the estimates

• Codex receipts report input, cached input, output and reasoning tokens per run; cached input is billed at the cache-read rate and the rest of the input at the standard rate. Cache writes are free on this API and are not modelled.

• Standard tier, short-context rates. Per-request context sizes are not exposed by the receipts, so no long-context premium is applied.

• No Batch, Flex, Fast or priority pricing is applied.

• The estimate covers the solver's exposed receipts, not an independently observed API invoice.

• Max covers 22 runs: paddockcore never started because the Codex subscription quota window closed until September 19, 2026; it will be run and judged as a top-up.

• The sweep stamped each run at run time with a stale Terra list price ($2.50 input, $0.20 cached, $15 output); the frozen population record carries those stamps (frozen_record_usd per run). Every run was re-priced from its receipt at the 2026-09-11 list on 2026-09-16, and estimated_usd is the re-priced value.

Evidence and reproduction

Public files: `runs.json` and `runs.csv` with every run's factors, both judges' six-dimension sub-scores and intent recovery; `groups.json`; `calibration.json` with every gate value and control mean; `judge-protocols.json` with the exact rubric, system text, probe and match instructions, schemas and weights; `economics.json` with cost and token aggregates, the rate table and pricing limitations; and `provenance.json` with source hashes.
https://github.com/morganlinton/VulcanBenchCOM/tree/main/assets/data/swe-v4-terra-v36

Withheld: raw prompts and responses; submitted patches and reconstructed sources; quirk answer keys (they describe hidden-test behaviour); reviewer session identifiers and usage receipts.

Recompute the published numbers

Each run's Code quality is (0.24 x reviewed score + 0.09 x intent recovery) / 0.33, where both terms are the equal mean of the two judges, or the reviewed score alone where the submission passed no quirk family. The combined score is 100 x (0.50 F + 0.085 Q + 0.085 S + 0.33 C / 100) with F, Q, S on a 0 to 1 scale. Group means weight the cell's tasks equally. The export script recomputes every row from the frozen summary, re-prices every run from its receipt, and refuses to write if any value differs.

What this report does not claim

No human rated anything; the scores are model judgment for a human reader, not human validation. Standard errors describe task sampling only. The measured-maintenance layer is unbuilt, so the 33 points are 24 reviewed plus 9 intent recovery. Runs were not repeated and effort labels are Codex's own. The Max cell holds 22 runs until paddockcore can run and is judged as a top-up. Hash checks bind the export to frozen files; they do not prove that judges were unbiased or that no training overlap exists.

Operator record

Each judge made 437 calls: 80 in calibration, 114 primary reviews, 5 repeats, 10 pairwise checks, 114 intent probes and 114 answer-key matches. Muse needed a second attempt on six calls, all for an unsupported excerpt; on one of them both attempts quoted the same line with its whitespace collapsed, and the first response was selected with the excerpt re-wrapped to the source under the standing recovery rule. Grok needed a second attempt on eight: five unsupported excerpts, one match that cited a departure not on its own list, and two transport faults when the Cursor CLI could not resolve its API host, each retried under the network-fault rule. Neither judge produced a reviewer fallback. Every attempt is archived beside its replacement in the harness run directory.

Source hashes

Frozen artifact	SHA-256
v3.6/summary.json	e3ddd51feb26ec3a0808c83d7461ff04…
v3.6/protocol.json	f6214c5c99d75e61a5803811700cec86…
v3.6/private-manifest.json	7a094a6b4e063cd608b7c42ecf1090c3…
v3.6/calibration-muse.json	e5da52d79e6dc06761758e8892abea44…
v3.6/calibration-grok.json	df81d4be726cda10d303d64f7f7dab13…
comparison.json	5834b8e04f2cf92bc8bea24858ab3e6c…

The protocol documents, controls, quirk-key format, runner and operator wrapper are in the VulcanBench repository under docs/judging and harness. The quirk answer keys themselves describe hidden-test behaviour and are not published.

Appendix. Per task at Low and Max effort

Task	Low CQ	Max CQ	Low combined	Max combined
blendcore	70.5	66.1	61.17	87.28
cellarcore	68.2	60.6	41.14	84.82
codeccore	74.7	73.1	65.13	89.29
depotcore	60.0	54.1	66.89	82.55
freightcore	82.5	78.8	81.45	91.28
granarycore	83.3	73.6	42.26	88.85
hedgecore	64.8	81.5	68.16	92.29
lodgecore	70.5	70.5	50.78	87.94
matchcore-order-book	87.9	78.0	44.30	90.92
metercore	73.6	79.0	89.71	91.29
pacecore	73.5	87.1	39.84	94.26
paddockcore	62.9	not run	35.05	not run
payrollcore	69.3	75.8	38.33	90.44
qlite-store	78.0	84.1	41.07	93.00
queuecore	75.0	61.3	59.03	85.31
quotacore	68.3	74.2	78.12	88.71
reflowcore	65.0	77.7	62.16	91.19
schedcore	67.0	71.0	49.67	88.47
settlecore	84.8	79.5	63.29	91.35
snapcore	74.8	74.1	65.34	90.02
stampcore	70.5	65.2	72.15	86.95
tallycore	63.3	70.9	67.67	88.87
vaultcore	74.9	82.7	90.39	92.99

Per-task values from runs.json. Task names are shortened for width. Paddockcore at Max waits for the Codex quota window.

VulcanBench

September 2026

VulcanBench-SWE v4: GPT-5.6 Terra across effort levels

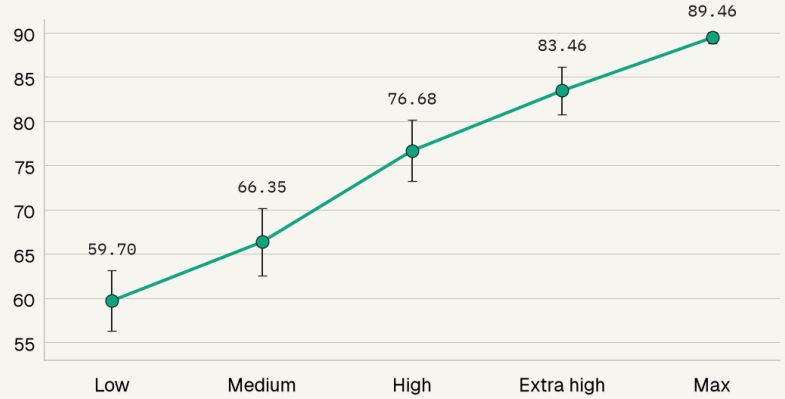
Combined score and runtime at every effort level. 23 tasks per effort, 22 at max. Code quality judged by Muse Spark 1.3 and Grok 4.6.

● **GPT-5.6 Terra · Codex**

n=23 at every effort, n=22 at max

Combined score

/100 · higher is better · focused scale



Mean runtime

Minutes per task · lower is better

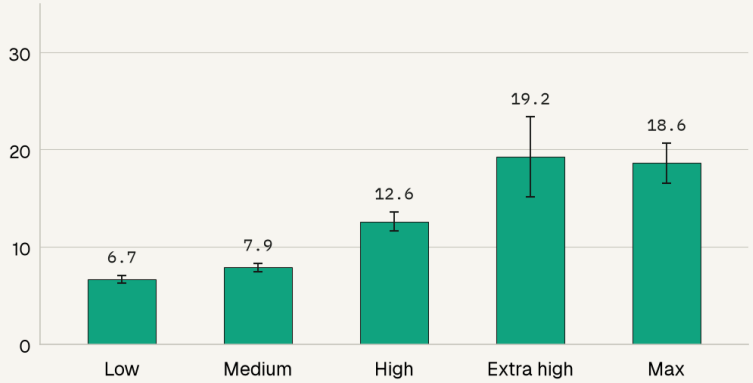


Table 1 | Code quality at each effort level

Component	Low	Medium	High	Extra high	Max
					/100, mean of both judges
Code quality	**72.31**	**70.93**	**73.90**	**73.13**	**73.58**
Human readability	68.3	67.1	70.0	67.9	68.0
Maintainability	73.2	71.0	74.1	74.3	73.9
Intent recovery	76.2	75.9	78.7	78.5	80.7
Rated by Muse Spark 1.3	69.4	67.9	71.9	70.9	71.1
Rated by Grok 4.6	72.1	70.2	72.2	71.3	70.7
Standard error of Code quality	1.54	1.54	1.50	1.76	1.76

VulcanBench

September 2026

VulcanBench-SWE v4: GPT-5.6 Terra across effort levels

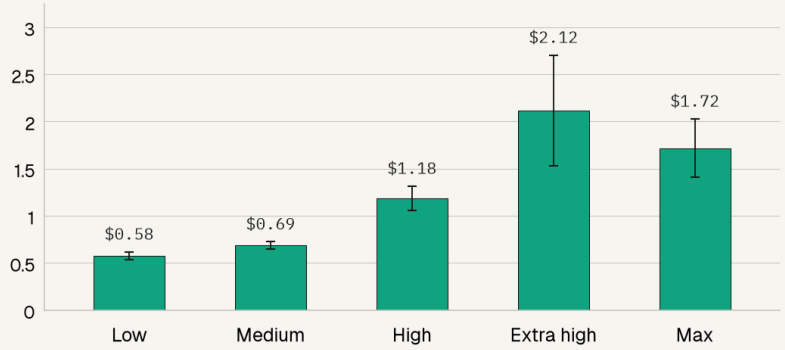
API-equivalent cost and token use at every effort level. Same 114 runs as the score card; solver inference only, judging excluded.

● **GPT-5.6 Terra · Codex**

n=23 at every effort, n=22 at max

API-equivalent cost

USD per task at list prices · lower is better



Tokens per task

Millions of raw tokens, cache reads included · lower is better

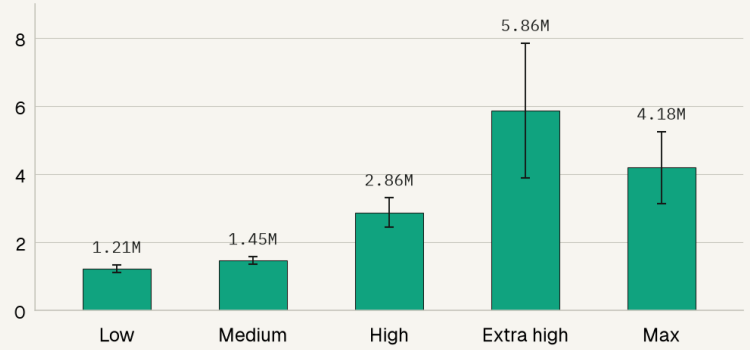


Table 1 | Cost, tokens and runtime at each effort level

Effort	Runs	$/task	Level total API-equivalent, list prices	Tokens/task raw, millions	Minutes/task solver wall-clock
Low	23	$0.58	$13.26	1.21M	6.7
Medium	23	$0.69	$15.79	1.45M	7.9
High	23	$1.18	$27.22	2.86M	12.6
Extra high	23	$2.12	$48.71	5.86M	19.2
Max	22	$1.72	$37.76	4.18M	18.6
Full sweep	**114**	**$1.25**	**$142.73**	**354M**	**24.6 h**

USD at list prices checked 2026-09-11, cache-aware (Terra $2.00 input, $0.20 cached input, $12.00 output per million), solver inference only; subscription bills differ.

Max has 22 runs: paddockcore never started because the Codex subscription quota window closed until September 19; it will be judged as a top-up.

Whiskers are one task standard error. Tokens are raw solver totals including cache reads.