GPT-5.5 vs. GPT-5.6 Luna across every effort level

VulcanBench-SWE v4 | Code quality protocol v3.5 | September 2026

Abstract

GPT-5.5 and GPT-5.6 Luna ran the 23-task VulcanBench-SWE v4 suite through the Codex CLI on a ChatGPT subscription, once per task at every effort level each API accepts: four levels for GPT-5.5 and five for Luna, 207 runs in all. 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 Astra and Fable 5.1 report, with a ground-truth intent-recovery probe. The effort knob decides the comparison: GPT-5.5 leads at Low and Medium by 8.52 and 10.26 points, Luna leads at High and Extra-high by 1.29 and 0.66 points, inside one standard error, and Luna's Max level, which GPT-5.5 does not offer, is the top cell at 84.15 with 19 of 23 tasks passed. Luna is rated higher on Code quality at every matched effort, by under three points from Medium up. At list API rates Luna costs 16.6 to 74.8 times less per task at matched effort.

Under the prior 50/15/15/20 profile applied to the same Code quality scores the same ordering holds at every effort, 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
GPT-5.5	Low	49.80	2.93	53.44	61.75	8.2	1/23
GPT-5.5	Medium	63.54	3.84	66.78	64.65	12.4	3/23
GPT-5.5	High	69.92	3.81	73.32	62.93	18.2	7/23
GPT-5.5	Extra-high	78.46	2.51	81.25	67.67	20.8	11/23
Luna	Low	41.29	0.96	43.72	71.73	2.7	0/23
Luna	Medium	53.28	2.94	56.46	65.58	7.4	1/23
Luna	High	71.22	3.71	74.32	65.79	21.0	9/23
Luna	Extra-high	79.11	3.07	81.87	68.05	25.4	13/23
Luna	Max	84.15	1.35	87.18	65.35	44.1	19/23

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 23 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. GPT-5.5's API has no Max level.

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 v3.5 is the v3.4 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 v3.5 before any counted call, and both neutral judges are scored in one directory with no sensitivity panels.

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 9 Luna Low runs and 1 Luna Medium 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-v3.5	80	passed	yes, g11 by 0.02	g11_repeatability
Grok 4.6	code-quality-maintenance-v3.5	80	passed	no	none

Muse Spark 1.3 used the pre-registered allowance: 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	4.00	3.80	3.40	4.00	4.00	3.80
Muse	compressed	1.10	1.80	1.00	3.70	2.60	3.10
Muse	formatted	1.00	3.60	1.40	4.00	2.90	3.10
Muse	misleading comments	4.00	3.90	0.20	4.00	2.60	3.50
Muse	hidden state	2.60	3.10	2.00	1.50	2.50	1.70
Grok	clear	3.90	3.50	3.20	4.00	4.00	3.50
Grok	compressed	1.60	1.60	1.40	3.20	2.60	2.90
Grok	formatted	1.50	2.30	1.70	3.20	3.00	3.00
Grok	misleading comments	3.60	3.30	1.00	3.60	2.10	3.10
Grok	hidden state	3.00	2.90	2.50	2.00	3.30	2.40

Both judges separate the clear program from the compressed one by more than two points on naming and about two points on presentation, credit formatting on presentation but not on naming, 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 model's best effort

Component	GPT-5.5, Extra-high	Luna, Max	Luna minus GPT-5.5
Code quality	67.67	65.35	-2.32
Human readability	59.6	55.8	-3.8
Maintainability	69.5	64.3	-5.2
Intent recovery	76.0	79.5	+3.5
Reviewed score	64.5	60.1	-4.5
Rated by Muse Spark 1.3	62.8	58.0	-4.8
Rated by Grok 4.6	66.3	62.1	-4.2
Standard error of Code quality	1.49	1.33	

At each model's best effort the reviewed scores are close and GPT-5.5 reads slightly better; Luna recovers more of the hidden contract. Across matched efforts Luna's Code quality and human readability are higher at every level, while maintainability and intent recovery trade places; both judges rank the reviewed score the same way in every cell. Both models sit well below the Astra and Fable 5.1 board under the same rubric, 62 to 72 here against 70 to 82 there.

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

Model	Effort	Functional	Lint, complexity	Security	Code quality
GPT-5.5	Low	28.35	79.39	100.00	61.75
GPT-5.5	Medium	53.96	79.13	100.00	64.65
GPT-5.5	High	68.02	78.37	99.78	62.93
GPT-5.5	Extra-high	81.93	78.36	100.00	67.67
Luna	Low	4.49	80.85	100.00	71.73
Luna	Medium	32.65	80.12	100.00	65.58
Luna	High	68.53	79.31	100.00	65.79
Luna	Extra-high	82.97	78.52	100.00	68.05
Luna	Max	95.00	78.08	99.35	65.35

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 207 runs priced from their Codex receipts at list API rates, cache-aware, solver inference only, judging excluded. Both models ran on a ChatGPT subscription, so these are API-equivalent estimates rather than bills. GPT-5.5 lists at 25 times Luna's input, cached-input and output rates. Luna uses more tokens than GPT-5.5 from High up, most of them cache reads.

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

Effort	GPT-5.5 $/task	Luna $/task	Ratio	GPT-5.5 tokens	Luna tokens	GPT-5.5 min	Luna min
Low	$1.79	$0.02	74.8x	1.72M	0.42M	8.2	2.7
Medium	$2.71	$0.07	40.8x	2.74M	1.35M	12.4	7.4
High	$4.20	$0.22	19.4x	4.52M	5.89M	18.2	21.0
Extra-high	$4.56	$0.27	16.6x	4.72M	7.58M	20.8	25.4
Max	no level	$0.45			11.88M		44.1
Full sweeps	$305.04	$23.70	12.9x	315M	624M	22.9 h	38.6 h

Rates checked 2026-09-11: GPT-5.5 at $5.00 input, $0.50 cached input and $30.00 output per million tokens; Luna at $0.20, $0.02 and $1.20. 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 full sweeps are 92 GPT-5.5 runs and 115 Luna runs. 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.

• One GPT-5.5 extra-high run (paddockcore) was re-run after the first attempt overran its cap under a harness fault; the re-run is the priced and judged run and the capped attempt is excluded.

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-gpt55-luna-v35

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 23 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. One GPT-5.5 Extra-high task was re-run after its first attempt overran the cap under a harness fault; the re-run is the judged run. 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 718 calls: 80 in calibration, 207 primary reviews, 9 repeats, 8 pairwise checks, 207 intent probes and 207 answer-key matches. Muse needed a second attempt on five calls (two unsupported excerpts, two malformed JSON responses, one missing consequence in calibration) and Grok on six (four unsupported excerpts, one malformed response, and one transport fault when the Cursor CLI could not resolve its API host). The transport fault stopped the run for a person; the operator wrapper gained a rule that gives one fresh attempt after a judge CLI network fault, with the receipt retained, and the call was retried under it. Neither judge produced a reviewer fallback. Every second attempt is archived beside the first in the harness run directory.

Source hashes

Frozen artifact	SHA-256
v3.5/summary.json	faea0a8e7a99c02aa18e311e0340111f…
v3.5/protocol.json	e2c2afdb5149cd3999ce10b79daa7830…
v3.5/private-manifest.json	1b689049006131e96770b952f4f598da…
v3.5/calibration-muse.json	a88aed191f3db3b10fb73341e12aa2e6…
v3.5/calibration-grok.json	fd69ce31e8338ea68222158364df790e…
comparison.json	cc114b736cbdad27595c1a9154ab042a…

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 each model's best effort

Task	GPT-5.5 CQ	Luna CQ	Difference	GPT-5.5 combined	Luna combined
blendcore	67.8	61.5	-6.3	87.64	85.68
cellarcore	63.3	62.9	-0.3	71.43	85.56
codeccore	75.4	71.0	-4.4	70.22	78.56
depotcore	59.6	55.0	-4.7	62.95	82.71
freightcore	77.3	73.5	-3.8	90.84	89.59
granarycore	61.4	50.5	-10.9	84.27	80.78
hedgecore	73.4	72.5	-0.9	89.62	89.30
lodgecore	71.3	65.6	-5.8	88.15	86.38
matchcore-order-book	65.9	65.9	+0.0	86.73	86.55
metercore	72.6	70.6	-2.0	89.44	88.65
pacecore	66.3	73.1	+6.8	81.71	89.52
paddockcore	52.1	52.9	+0.8	58.26	80.67
payrollcore	68.9	63.6	-5.3	83.02	86.22
qlite-store	75.8	67.4	-8.3	90.20	87.43
queuecore	60.6	63.6	+3.0	72.85	73.76
quotacore	69.1	72.9	+3.8	78.30	79.63
reflowcore	68.3	60.2	-8.1	87.85	85.42
schedcore	67.9	72.3	+4.4	81.31	88.83
settlecore	74.7	64.1	-10.6	64.85	60.98
snapcore	58.7	67.4	+8.7	64.81	87.76
stampcore	68.8	66.9	-1.9	88.16	87.57
tallycore	56.8	66.6	+9.7	84.14	87.52
vaultcore	80.3	63.1	-17.2	47.75	86.30

Per-task values from runs.json; GPT-5.5 at Extra-high and Luna at Max. Task names are shortened for width.

VulcanBench

September 2026

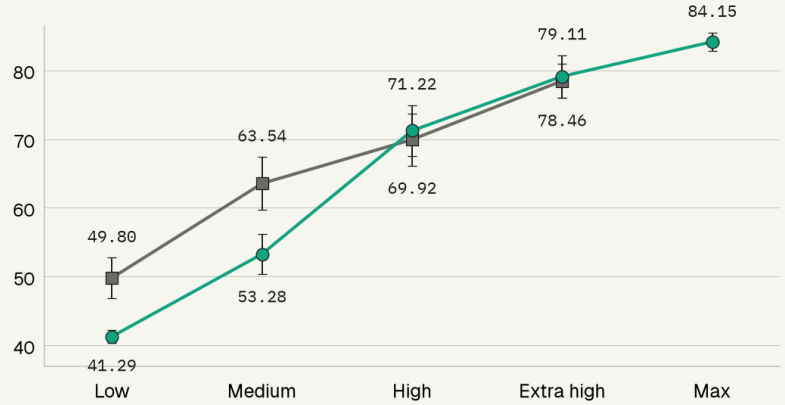
VulcanBench-SWE v4: GPT-5.5 vs. GPT-5.6 Luna

Combined score and runtime at every effort level. 23 tasks per model per effort; GPT-5.5 has no max level. Code quality judged by Muse Spark 1.3 and Grok 4.6.

■ GPT-5.5 · Codex ● GPT-5.6 Luna · Codex n=23 at every effort

Combined score

/100 · higher is better · focused scale



Mean runtime

Minutes per task · lower is better

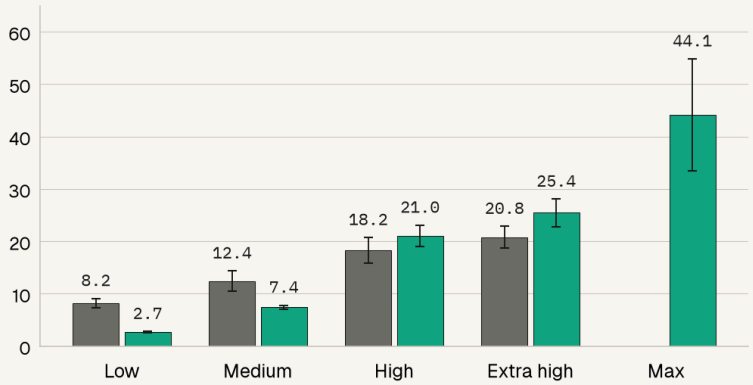


Table 1 | Code quality at each model's best effort

Component	GPT-5.5, extra high	Luna, max	Difference Luna minus GPT-5.5
Code quality	**67.67**	**65.35**	**-2.32**
Human readability	59.6	55.8	-3.8
Maintainability	69.5	64.3	-5.2
Intent recovery	76.0	79.5	+3.5
Rated by Muse Spark 1.3	62.8	58.0	-4.8
Rated by Grok 4.6	66.3	62.1	-4.2
Standard error of Code quality	1.49	1.33	

VulcanBench

September 2026

VulcanBench-SWE v4: GPT-5.5 vs. GPT-5.6 Luna

API-equivalent cost and token use at every effort level. Same 207 runs as the score card; solver inference only, judging excluded.

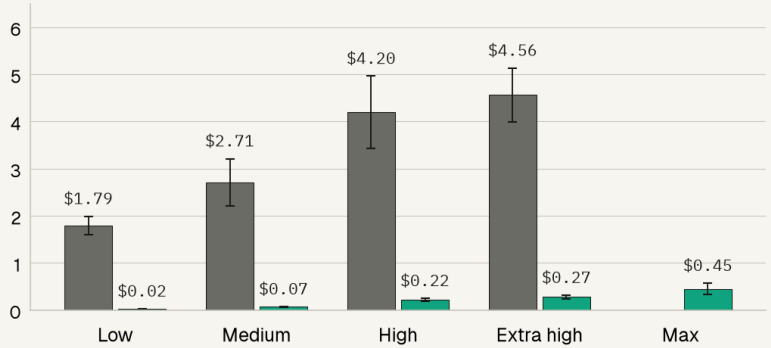
■ **GPT-5.5 · Codex**

● **GPT-5.6 Luna · Codex**

n=23 at every effort

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

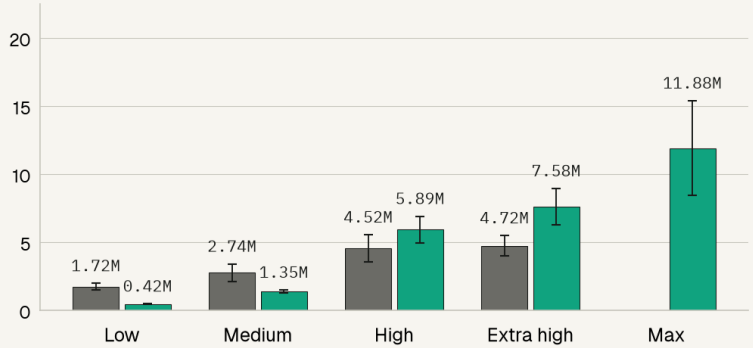


Table 1 | Cost and tokens at each effort level

Effort	GPT-5.5 $/task	Luna $/task API-equivalent, list prices	GPT-5.5 ÷ Luna cost ratio	GPT-5.5 tokens	Luna tokens raw tokens per task, millions
Low	$1.79	$0.02	74.8×	1.72M	0.42M
Medium	$2.71	$0.07	40.8×	2.74M	1.35M
High	$4.20	$0.22	19.4×	4.52M	5.89M
Extra high	$4.56	$0.27	16.6×	4.72M	7.58M
Max	no level	$0.45			11.88M
Full sweeps, 92 and 115 runs	**$305**	**$24**	**12.9×**	**315M**	**624M**

USD at list prices checked 2026-09-11, cache-aware (GPT-5.5 cached input $0.50 per million, Luna $0.02), solver inference only; subscription bills differ.

Solver time for the sweeps: GPT-5.5 22.9 h over four levels, Luna 38.6 h over five. GPT-5.5's API has no max level. One GPT-5.5 extra-high task was re-run after a harness fault; the capped first attempt is set aside.

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