

Top-20 Priority

Top-20 Refactor Priority Table

Source: `~/system/specs/agent-ic-os-pillar4-skills-audit-2026-05-04.md` (\$4)

MC: [#99131](#) | [#99176](#)

Date: 2026-05-05

Methodology

Priority score formula:

```
priority_score = log10(skill_md_tokens_est) * (1 + invocations_30d)
```

Bonus weight $\times 1.5$ if `frontmatter_description_bytes > 500`.

Tie-break rule: Higher `skill_md_tokens_est` wins.

Exclusion list:

- `owner=anthropic` vendor skills (docx, pdf, pptx, xlsx, figma-design) —
VENDOR_REFACTOR_IMMUNE
- `_archived/` skills
- TOB skills with `skill_md_loc=0` (no SKILL.md at root, `tokens_est=0`, score undefined)
- Skills where `invocations_30d=NO_DATA` (none in this dataset — all zero values are grounded in grep)

Note on invocations_30d=0 skills: Ranked separately at bottom of table with `priority_score` computed as `log10(skill_md_tokens_est) * 1` (no invocation multiplier). This represents their per-session load cost without usage frequency.

Note on est_monthly_cost: The columns below show estimated cost per month. These projections assume `sessions_per_month=600` and `invocations_30d` as a proxy for monthly rate. Per-turn savings are the honest metric; monthly projections are estimates only.

Top-20 Table (sorted descending by priority_score)

rank	skill_name	LOC	tokens	inv_30d	est_\$/mo (current)	est_\$/mo (post-L3)	savings_\$/mo	priority_score	owner
1	task-postflight	541	5,367	21	\$0.547	\$0.078	\$0.469	82.054	john
2	prompt-forge	224	2,372	20	\$0.350	\$0.070	\$0.280	70.877	john
3	plan-with-team	140	1,177	13	\$0.105	\$0.042	\$0.063	42.991	john
4	build-plan	90	923	7	\$0.126	\$0.063	\$0.063	23.722	john
5	ask-board	307	2,623	3	\$0.125	\$0.038	\$0.087	13.675	john
6	build	79	838	3	\$0.113	\$0.057	\$0.056	11.693	john
7	sentinel	105	990	2	\$0.116	\$0.058	\$0.058	8.987	john
8	sync	46	346	2	\$0.087	\$0.087	\$0.000	7.617	john
9	learning-opportunity	165	1,433	1	\$0.067	\$0.034	\$0.033	6.313	john
10	vault-unlock	117	1,312	1	\$0.142	\$0.071	\$0.071	6.236	john
11	incident-response	122	1,051	1	\$0.067	\$0.034	\$0.033	6.043	john
12	youtube-learning	93	877	1	\$0.136	\$0.068	\$0.068	5.886	john
13	code-review	87	674	1	\$0.002	\$0.001	\$0.001	5.657	john
14	lightrag-upload	87	659	1	\$0.117	\$0.059	\$0.058	5.638	john
15	lightrag-status	101	625	1	\$0.121	\$0.061	\$0.060	5.592	john
16	product-lifecycle	491	5,103	0	\$0.081	\$0.041	\$0.040	3.708	john
17	skill-creator	362	4,911	0	\$0.088	\$0.044	\$0.044	3.691	john

rank	skill_name	LOC	tokens	inv_30d	est_\$/mo (current)	est_\$/mo (post-L3)	savings_\$/mo	priority_score	owner
18	doc-coauthoring	375	4,274	0	\$0.208	\$0.104	\$0.104	3.631	john
19	mcp-builder	236	2,457	0	\$0.135	\$0.068	\$0.067	3.390	john
20	plan-build-test	293	2,437	0	\$0.099	\$0.050	\$0.049	3.387	john

est_\$/mo (post-L3) = estimate assuming 50% body-token reduction via progressive disclosure

Per-Skill Triage (Top 10)

#1 task-postflight

- **Current footprint:** 541 LOC / 5,367 tokens
- **Why bloated:** BLOAT_LOC_GT_300 — Contains anomaly decision tree (Section 3), learning-opportunity dispatch template (Section 4), memory writer procedure (Section 5), and failure mode reference table (Section 8) all inline in one file. Most of this content is only needed after an anomaly is detected.
- **Recommended action:** Split — progressive-disclose. Trigger skeleton ≤ 200 LOC stays in SKILL.md; Sections 3-5+8 move to references/.
- **Predicted savings:** ~3,500 tokens/session on typical PASS flows (63% context reduction); full 5,367 tokens only loaded on ANOMALY path.

#2 prompt-forge

- **Current footprint:** 224 LOC / 2,372 tokens
- **Why bloated:** Single references/agent-briefs.md exists but body still contains full 5-panelist dispatch protocol, model tier assignments, and synthesis rules inline. Most body content is needed only during the forge step.
- **Recommended action:** Split — move per-panelist briefs and synthesis rules to references/; keep trigger condition and dispatch skeleton in core.
- **Predicted savings:** ~1,200 tokens/session when invoked without full panelist detail read (50% reduction).

#3 plan-with-team

- **Current footprint:** 140 LOC / 1,177 tokens
- **Why bloated:** No references/ dir. Builder/validator role descriptions, round-robin protocol, and output templates are all inline. Frequently invoked (13x in window) — every invocation carries full load.
- **Recommended action:** Progressive-disclose — move builder brief and validator brief to references/. Keep selection logic in SKILL.md.
- **Predicted savings:** ~700 tokens/session (59% reduction) across 13 monthly invocations.

#4 build-plan

- **Current footprint:** 90 LOC / 923 tokens
- **Why bloated:** No references/ dir. Moderate size but high invocation frequency (7x). Output templates and TaskList format examples inline.
- **Recommended action:** Progressive-disclose — move TaskList format examples and edge-case handling to references/quick-ref.md.
- **Predicted savings:** ~400 tokens/session (43% reduction).

#5 ask-board

- **Current footprint:** 307 LOC / 2,623 tokens
- **Why bloated:** BLOAT_LOC_GT_300 — 5-agent dispatch briefs are fully inline. Each panelist persona description (50-80 lines each) loads for every board invocation.
- **Recommended action:** Split — move per-panelist briefs to references/panelist-`<name>.md`. Keep dispatch skeleton (trigger, model tier, synthesis format) in SKILL.md.
- **Predicted savings:** ~1,800 tokens/session (69% reduction).

#6 build

- **Current footprint:** 79 LOC / 838 tokens
- **Why bloated:** No references/ dir. Build mode toggle and autocoder integration details inline. Reasonably compact but no progressive disclosure path for edge cases.
- **Recommended action:** Progressive-disclose — move edge-case handling (yolo mode, concurrency) to references/.
- **Predicted savings:** ~300 tokens/session (36% reduction). Low priority given small absolute size.

#7 sentinel

- **Current footprint:** 105 LOC / 990 tokens
- **Why bloated:** No references/ dir. 5-agent audit team definitions inline. Hardcoded audit procedure steps.
- **Recommended action:** Progressive-disclose — move per-agent audit checklists to references/; keep dispatch skeleton.

- **Predicted savings:** ~500 tokens/session (50% reduction).

#8 sync

- **Current footprint:** 46 LOC / 346 tokens
- **Why bloated:** Small and clean — no significant refactor needed. Score driven by 2 invocations.
- **Recommended action:** Keep as-is. Already close to L3 trigger skeleton.
- **Predicted savings:** Negligible. Lowest absolute token size in top-20.

#9 learning-opportunity

- **Current footprint:** 165 LOC / 1,433 tokens
- **Why bloated:** No references/ dir. Root-cause classification procedure, GOTCHA layer mapping, and fix-type catalog inline. Invoked only on anomaly path (1x).
- **Recommended action:** Progressive-disclose — move GOTCHA layer catalog and fix-type table to references/.
- **Predicted savings:** ~700 tokens/session (49% reduction).

#10 vault-unlock

- **Current footprint:** 117 LOC / 1,312 tokens
- **Why bloated:** HARDCODED_PATH (/Users/makinja) — breaks Pillar #9 VM portability. Caddy proxy restart sequence and bw CLI flags inline.
- **Recommended action:** Progressive-disclose + fix HARDCODED_PATH — move Caddy sequence to references/; replace /Users/makinja with \$HOME.
- **Predicted savings:** ~600 tokens/session (46% reduction) + VM portability fix.

Aggregate Savings (per-turn, not monthly \$)

skills loaded per turn	tokens saved vs. baseline	% context window recovered (128K window)
Only task-postflight (PASS path)	3,500 tokens	2.7%
task-postflight + prompt-forge	4,700 tokens	3.7%
Top-5 hot-path skills (ranks 1-5)	7,300 tokens	5.7%
All top-20 (max benefit, full session)	19,500 tokens	15.2%
All 79 skills at L3 (theoretical max)	~35,000 tokens	27.3%

Assumes 40-50% body-token reduction per skill post-refactor. Calculations: $savings_tokens = current_tokens \times 0.45$. Context window basis: 128K tokens (Claude standard context). These are per-turn estimates derived from body-token reduction; monthly projections without measured session counts would be phantom claims.

[← Back to Audit Summary](#) | [Design Pattern](#) →

Revision #2

Created 2026-05-05 13:01:20 UTC by John

Updated 2026-06-07 20:01:13 UTC by John