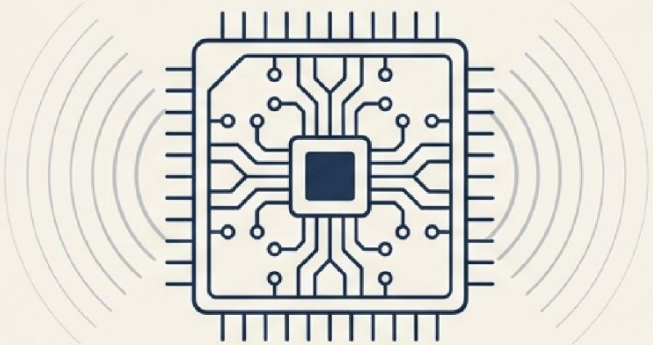
Bridging the Procedural Gap with AI Agent Skills

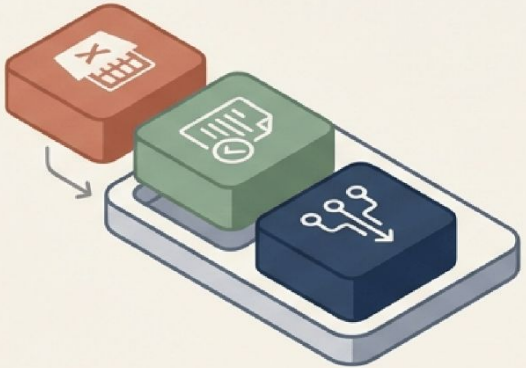
Based on an analysis of 7,308 agent trajectories across 84 professional tasks.

Foundation models possess general reasoning but lack procedural knowledge



Foundation Models = The CPU

Excel at broad reasoning and general corpora synthesis.



Agent Skills = The Software Apps

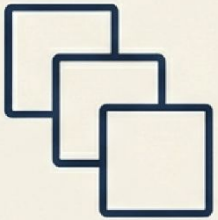
Provide the exact Excel formulas, regulatory formats, or data pipelines required for real-world work.

An agent may understand that a financial analysis requires calculating weighted averages, but it lacks the exact regulatory reporting conventions practitioners use daily.

Four criteria define an effective Agent Skill

Procedural Content

Encodes how-to guidance and standard operating procedures (e.g., USGS flood frequency analysis), not just factual retrieval.



Task-Class Applicability

Targets broad classes of problems (e.g., general financial valuation), not single-instance examples.

Structured Components

Bundles a required SKILL.md with optional executable scripts, code templates, and reference documentation.

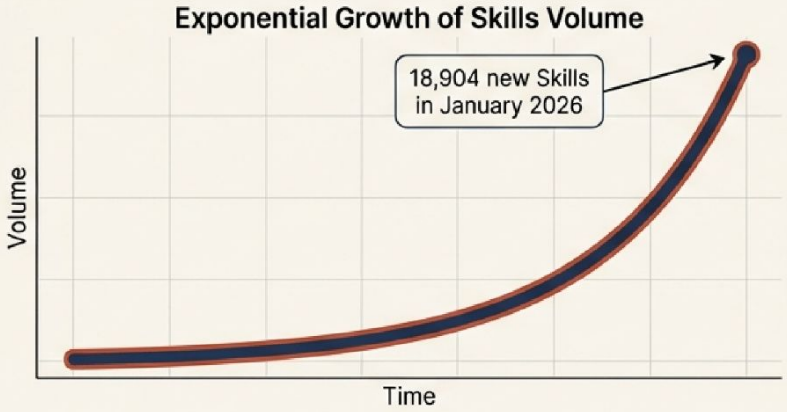
Portability

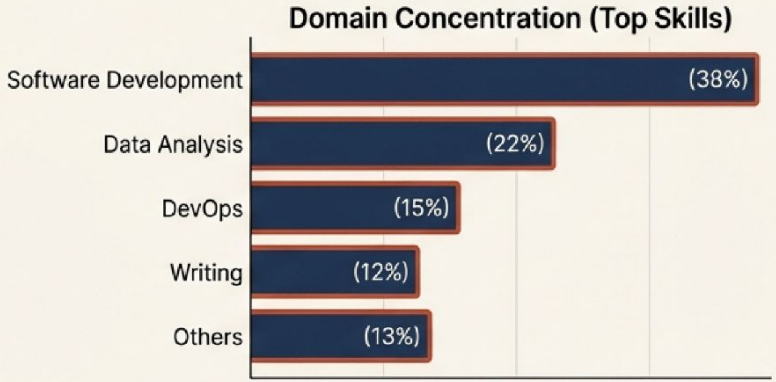
Functions as a version-controlled file-system artifact, independent of specific agent harnesses.

Skills occupy a distinct architectural position in the augmentation space

Retrieval-Augmented Generation (RAG)	Agent Skills
Retrieves declarative facts.	Skills package complete procedural workflows.
Tool Use	Agent Skills
Extends agent actions.	Skills guide the orchestration of those actions.
Fine-Tuning	Agent Skills
Modifies foundational model weights at high cost.	Skills preserve model generality while adding domain competence at inference time.

The ecosystem is expanding exponentially, but quality remains highly variable





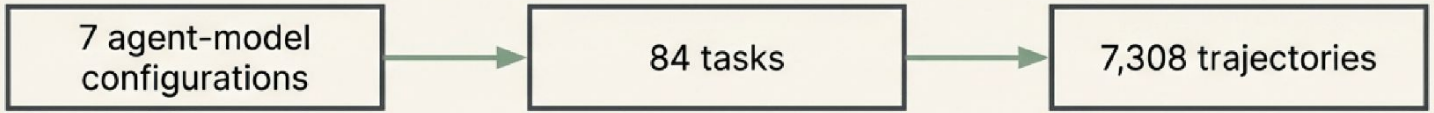
- **47,150** unique Skills identified across community and corporate repositories.
- **Median size** is lightweight (1,569 tokens / ~2.3 KB), mostly markdown files.
- **The Quality Gap:** The ecosystem mean quality score is only 6.2 out of 12, indicating a severe lack of standardized authoring and validation practices.

6.2 / 12

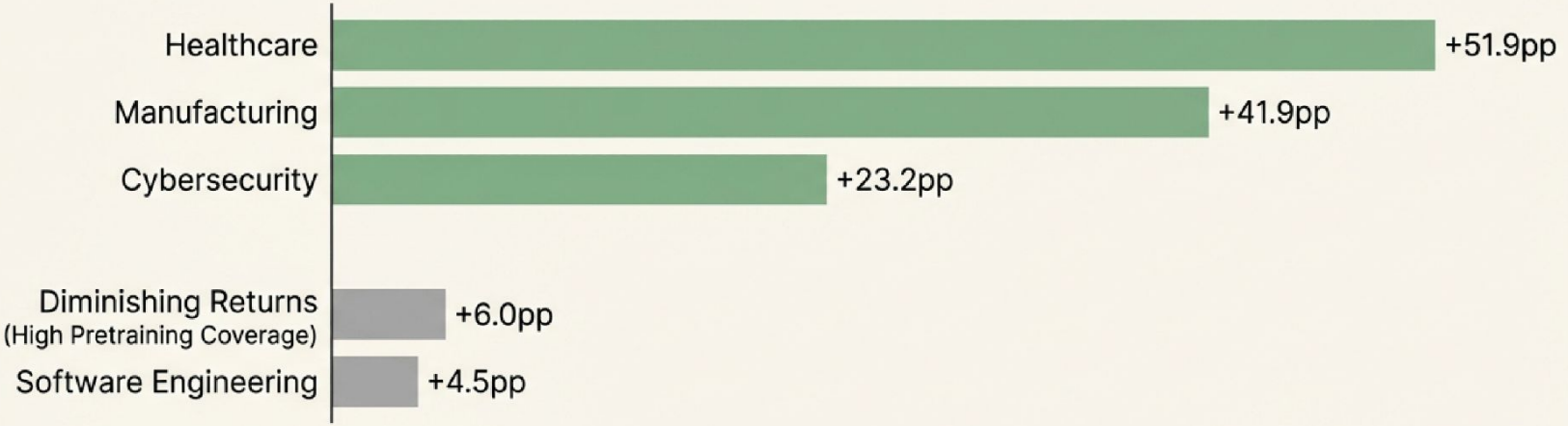
Curated Skills improve task completion rates by an average of 16.2 percentage points

+16.2pp

Under rigorous benchmarking, injecting structured procedural knowledge at inference time yields a massive aggregate improvement compared to baseline context augmentation.

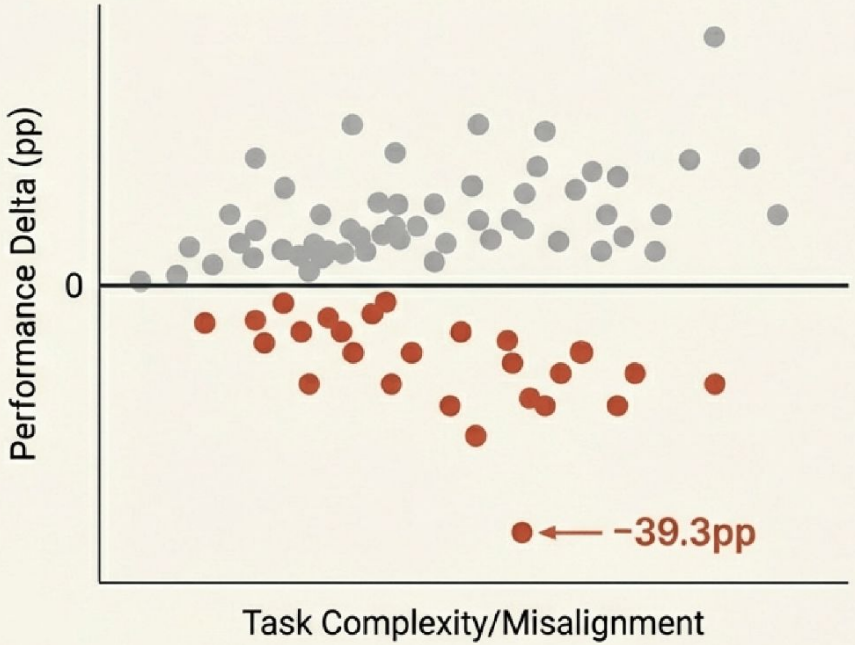


Skills efficacy is inversely correlated with foundation model pretraining coverage



Takeaway: Models benefit most from augmentation in domains requiring highly specialized, proprietary, or procedurally rigid workflows that do not exist in web-scale data.

Augmentation introduces overhead when misaligned with task requirements



Warning

16 of 84 tasks saw negative performance with Skills.

Notable Failures: taxonomy-tree-merge (−39.3pp), energy-ac-optimal-power-flow (−14.3pp).

The Cause: Verbose documentation, conflicting guidance, or introducing unnecessary complexity to tasks the base model already handles well.

Playbook 1: Target development toward procedurally complex, low-coverage domains

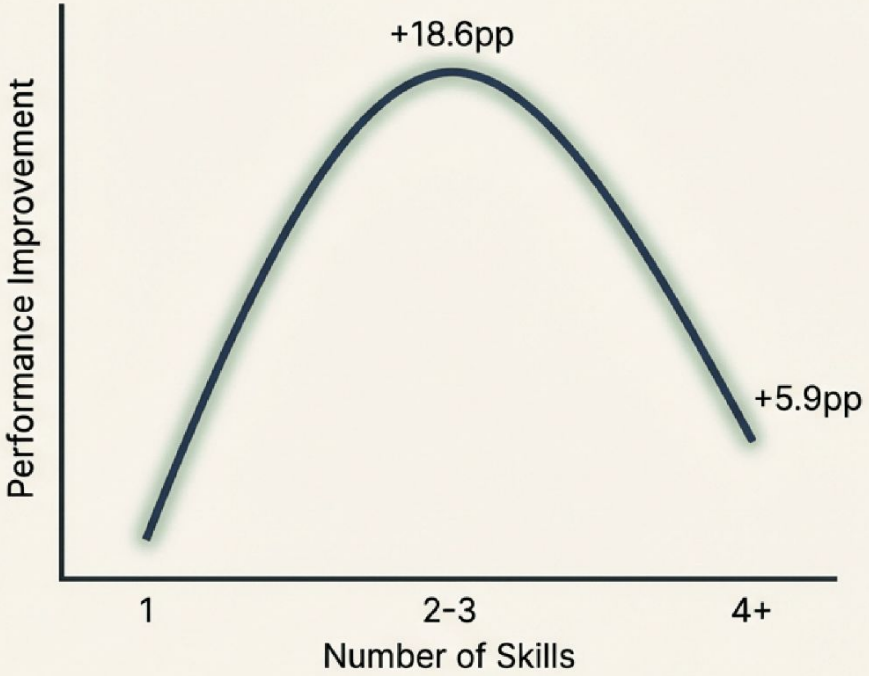
Eli Lilly

Encoded FDA regulatory submission requirements and protocol-specific data collection. Resulted in a 40% reduction in turnaround time for routine protocol amendments.

Tesla

Created Skills for production line optimization, capturing physics and factory constraints (e.g., constraint-based scheduling heuristics). Transformed tacit engineering knowledge into portable, version-controlled artifacts.

Playbook 2: Optimize for 2–3 modular Skills per task; avoid comprehensive bundles



- 2–3 Skills yields optimal improvement (**+18.6pp**). Providing 4+ Skills drops benefit to **+5.9pp**.
- Compact, actionable Skills (**+17.1pp**) vastly outperform comprehensive documentation (**-2.9pp**).

Stripe Case Study

Constrained API integration workflows (webhook verification, idempotency) to 2-3 key procedures with working code examples, successfully preventing context-window overwhelm.

Playbook 3: Adapt integration strategies to harness-specific constraints

Claude Code

Native integration. Achieves highest uplift (+23.3pp with Opus 4.5). Relies on structured frontmatter for discovery.

Gemini CLI

Requires agents to proactively invoke an activate_skill tool. Demands clear relevance descriptions to prompt agent usage.

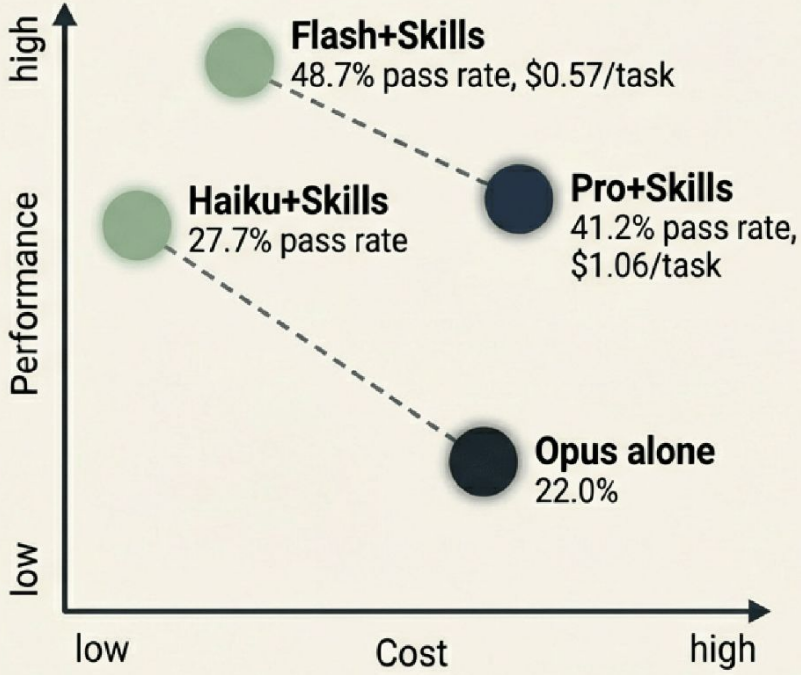
Codex CLI

Frequently acknowledges Skills but ignores them to write independent solutions. Requires embedding key procedures directly in task instructions.

Shopify Case Study

Improved cross-platform effectiveness by 12pp by maintaining harness-specific Skill variants.

Playbook 4: Use Skills to partially substitute for model scale, reducing inference costs

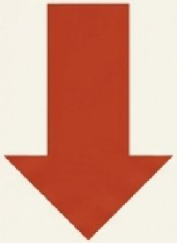


- Claude Haiku 4.5 + Skills (27.7% pass rate) outperforms Claude Opus 4.5 without Skills (22.0%).
- Gemini 3 Flash + Skills achieves 48.7% pass rate at $0.57/task, beating Gemini 3 Pro + Skills (41.2% at $1.06/task)—delivering superior performance for 47% less cost.

Salesforce Case Study

Deployed a tiered agent strategy. Routine tasks use Haiku + Skills, reserving Opus for novel scenarios. Reduced inference costs by **60%**.

Playbook 5: Avoid relying on agent autonomy for generating procedural knowledge



-1.3pp average drop

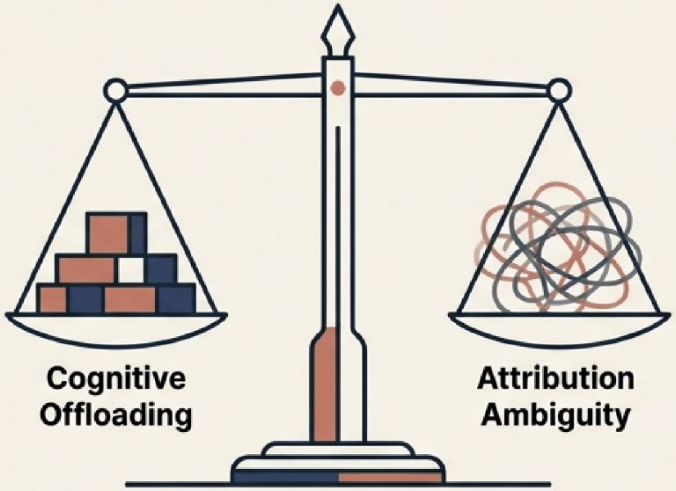
- **Self-generated Skills** provide a **−1.3pp average drop** in performance compared to baseline. Codex + GPT-5.2 drops by −5.6pp.
- **Failure Modes**: Models generate imprecise procedures (e.g., use pandas without specific API patterns) or fail to recognize specialization needs entirely.

GitHub Case Study

Attempted to use agent-generated Skills for repository operations to save time. Resulted in **severe underperformance** until pivoting to a **human-curated** community review model.

The user experience: Managing the transparency paradox and hidden labor

The Benefit: Reduces cognitive load for practitioners by offloading routine, procedurally complex workflows (reporting, data processing).



The Risk: Attribution ambiguity. When an agent fails despite having a Skill, users struggle to determine if the Skill was flawed, misapplied by the agent, or if the task was simply too hard.

The Hidden Cost: The ecosystem average quality of 6.2/12 proves that authoring reliable Skills requires domain expertise and iterative testing—effort that often goes unrecognized.

Establish systematic, automated validation before production deployment

1. Automated Validation

Structural checks, execution against oracle reference solutions, and leakage audits.

2. Human Review

Domain experts evaluate data validity, task realism, and procedural accuracy.

3. Acceptance Criteria

Strict deployment thresholds.

Capital One Case Study

Implemented this exact gating pipeline, requiring a ≥10pp improvement on at least one configuration. Reduced production Skill failures by 80%.

Implement feedback loops to capture failure patterns and drive iterative improvement

Skill exists, but the agent didn't retrieve it.

Agent retrieved the Skill, but implemented it incorrectly.

Agent followed the Skill perfectly, but produced incorrect output (flawed Skill).

New task patterns with no existing Skills.

Microsoft Case Study

Azure DevOps tracks these telemetry points to trigger expert revisions, increasing agent success rates by **25pp** over 12 months.

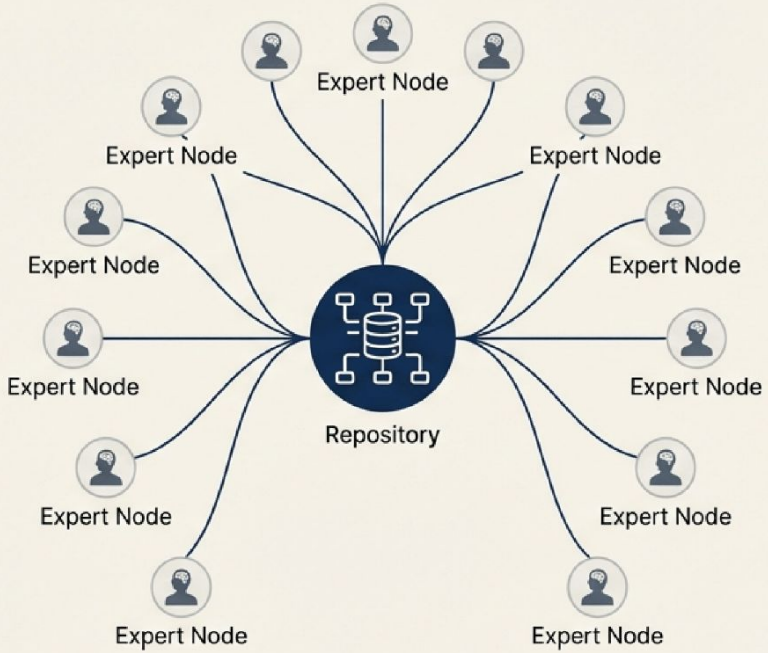
Scale coverage through distributed authoring and community contribution

Models consume knowledge effectively, but humans must author it.

Treat Skills authoring as legitimate technical work, not a volunteer side-project. Lower barriers with templates and automated validation feedback.

Stack Overflow Case Study

Created an open community contribution program with strict maintainer review based on completeness, clarity, specificity, and examples. Scaled coverage to 180+ specialized development workflows.



The Executive Blueprint for AI Agent Skills

1. **Target Strategically:** Invest in domains with complex procedures missing from general pretraining data (e.g., healthcare, manufacturing).
2. **Demand Modularity:** Enforce the Rule of 3. Optimize for compact, actionable procedures; ban comprehensive documentation bundles.
3. **Adapt to the Harness:** Customize Skill metadata and structure for Claude, Gemini, or Codex integration.
4. **Curate, Don't Automate:** Rely on human domain experts for authoring. Self-generated procedural knowledge actively degrades performance.
5. **Gate with Governance:** Implement automated validation and continuous telemetry to catch application failures and coverage gaps.