

AI research agents find well, verify poorly

How to build a reusable prompt for applied engineering research: what the evidence, practitioners, tools and published agent prompts say.

Anna Kubasiak · 2026-10-25 · researched with Claude (multi-agent deep research)

Evidence grades used throughout:

M independent measurement **V** vendor claim **P** practitioner / published practice **U** untested

Frontier research agents have largely solved *finding*: on BrowseComp, the hard single-fact web benchmark, OpenAI's Deep Research scored 51.5% against 29.2% for human trainers in 2025, and aggregator leaderboards put top systems near 90% by mid-2026. The measured failures sit downstream of the search. Agents attach citations that do not support the claim (12.5% to 97.5% unsupported statements depending on product), cite URLs that never existed (3-13%), pick SEO and aggregator sources, repeat stale API knowledge, and are badly calibrated (91% calibration error for Deep Research), so their stated confidence is worthless as a check. The interventions with measured effect are mostly *not* prompt wording. They are search effort, retriever quality, and an independent verification pass: a URL-checking tool cut dead citations to under 1%, and verifier-weighted sampling added up to 27 points on BrowseComp. On wording itself, controlled studies show two things to drop: expert personas, which give no accuracy gain, and chain-of-thought instructions, which add little for reasoning models. Almost everything else practitioners put in research prompts has not been tested. That covers clarifying questions, effort budgets, primary-source rules, quote-backed claims and anti-length rules, which rest on vendor guidance or on agreement between published agent prompts. So the reusable prompt should be a short, stable rule set inside a workflow: clarify and brief, tools with hard constraints set as parameters, budgets and stop rules, a claim ledger with quotes, a separate verification pass that uses a different method, a decision-ready output schema, and a gaps section. For engineering research, facts such as repo health, versions, licences and prices must come from live APIs and pages, not model recall. Where possible, test claims by building something.

What this report can vouch for: strong on citation audits, thin on engineering tasks and Reddit

Every recommendation below carries one of four evidence grades. These limits come first because they decide how far to trust each line.

GRADE	MEANING	TYPICAL SOURCES HERE
M	Independent measured study	arXiv/ICLR audits (DeepTRACE, BrowseComp-Plus, Rao et al.), Tow Center, persona and CoT studies
V	Vendor claim or vendor guidance, including vendor-run benchmarks	Anthropic engineering post and docs, OpenAI cookbook, Parallel's search benchmark
P	Practitioner anecdote, or convergent practice in published open-source prompts	HN threads, Willison, Steel's autopsy, the GPT Researcher/ODR/DeerFlow prompt files
U	Untested synthesis	This report's own inferences

The research behind this report has known holes. **Reddit was largely unreachable**: direct JSON access was blocked, and only mirror snapshots with near-zero engagement came through, so practitioner sentiment rests mostly on Hacker News and blogs. **No benchmark targets applied engineering research** (tool, library or vendor selection). Every quality figure comes from general web QA, PhD-style report tasks or news-citation tests. Most independent audits measured **2025-generation systems** (GPT-5 DR, Opus 4, Sonnet 4, Gemini 2.5). Current Claude 5.x, Gemini 3.x and GPT-5.x systems were not measured in any source read, so system rankings are dated. Several commercial products (Grok DeepSearch, Microsoft Copilot Researcher, Kimi, Manus, the You.com Research API, Perplexity consumer caps) rest **only on secondary sources**, and OpenAI's help centre returned 403. Model names that appear only in aggregators or in OpenAI cookbook notebooks (for example "GPT-5.6") are **not verified** against a vendor primary page. Search-API prices were read from live pricing pages on 2026-10-25 through a summarising fetch tool and should be spot-checked before anyone budgets with them. Not covered at all: prompt-injection risk from fetched content, deduplication features, paywall handling, and the Context7, Sourcegraph and Semantic Scholar/OpenAlex MCP servers.

Finding is nearly solved; citing, weighing and knowing when wrong are not

The benchmark curve is steep but narrow. BrowseComp asks 1,266 hard questions with short verifiable answers and by design excludes long-form or ambiguous tasks. In 2025, GPT-4o with browsing scored 1.9% and Deep Research 51.5% ([BrowseComp](#)). Aggregator snapshots from August-September 2026 cluster the top systems at **88-92%**, and the aggregator itself calls the benchmark near-saturated ([BenchLM](#))  /aggregator, unreplicated. Persistence in locating a fact is therefore no longer the bottleneck. The bottleneck is what happens between the fetched page and the sentence in the report. No saturated benchmark exists for that stage, and the audits that do exist are unflattering.

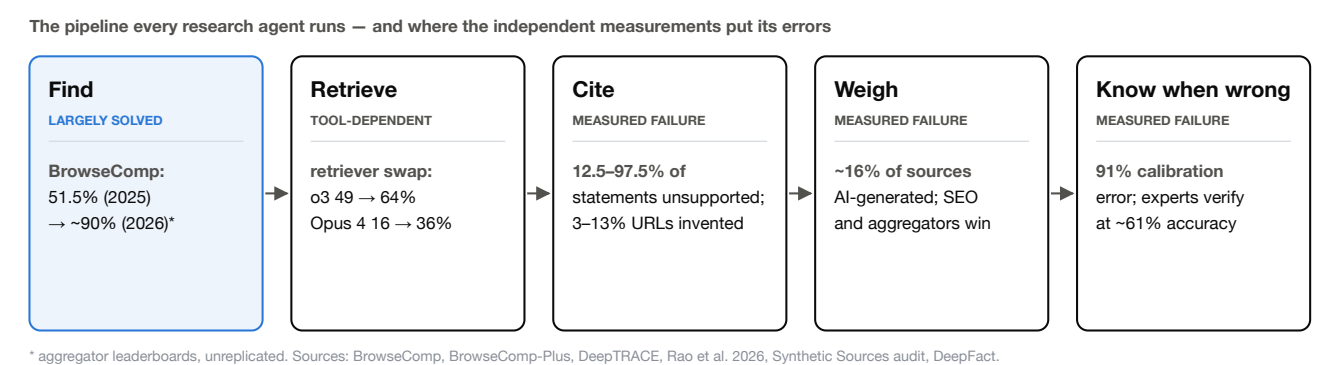


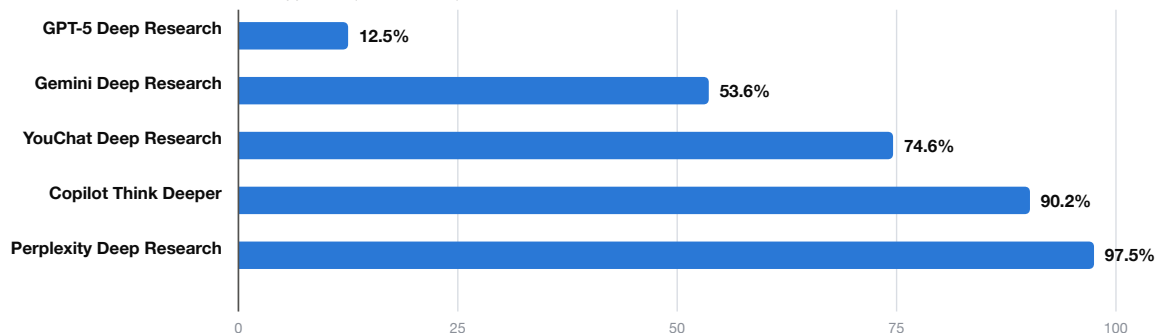
Figure 1. Finding is the stage benchmarks reward and the one that has improved; the measured failures sit in the stages after it.

FAILURE	MEASURED FIGURE	GRADE	SOURCE
Unsupported statements in deep-research reports	GPT-5 DR 12.5%, Gemini DR 53.6%, YouChat DR 74.6%, Copilot Think Deeper 90.2%, Perplexity DR 97.5%; citation accuracy 50-79%	 (judge-dependent, internal inconsistencies)	DeepTRACE
One-sided answers on debate questions	54.7% (GPT-5 DR) to 94.8% (Copilot TD)		DeepTRACE

FAILURE	MEASURED FIGURE	GRADE	SOURCE
Fabricated URLs	3-13% of cited URLs never existed; 5-18% fail to resolve; deep-research agents fabricate <i>more</i> than search-augmented LLMs	M	Rao, Wong, Callison-Burch 2026
Citation precision gap	o3 87-92% vs Opus 4 35-59%, Sonnet 4 32-52% on a fixed corpus	M (2025 models)	BrowseComp-Plus
Calibration	Deep Research 91% calibration error, worst of the models tested	V (OpenAI paper)	BrowseComp
Wrong source attribution	>60% wrong across 8 AI search tools; premium tiers more <i>confidently</i> wrong; ChatGPT hedged in 15/200 cases	M (Feb 2025, pre-deep-research)	Tow Center/CJR
Low-quality sources	~16% of cited sources show signs of AI generation	M (abstract-level)	Synthetic Sources audit
Stale technical knowledge	Code LLMs used deprecated APIs in 25-38% of completions	M (2024, code completion)	arXiv 2406.09834
Where agents break	Failures lie in evidence integration, verification and planning resilience, not task comprehension (14-mode taxonomy, ~1,000 reports)	M	FINDER/DEFT
Error propagation	Hallucinations propagate through plan, search and summarise stages, so checking only the final report misses them	M (abstract-only)	DeepHalluBench

Unsupported statements in deep-research reports

share of statements the cited sources do not support, % (lower is better)



Source: DeepTRACE (ICLR 2026). 2025-generation systems; LLM judge with moderate human agreement ($r = 0.62$).

Figure 2. The same audit, the same questions: an eightfold spread between products. System choice dominates.

Three conclusions follow. First, **system choice dominates "AI versus no AI"**: the same audit spans 12.5% to 97.5% unsupported statements. A research prompt cannot rescue a weak harness, and any ranking dates within months. Second, **citation volume is not support**. Deep-research modes cite more and fabricate more URLs at the same time ([Rao et al.](#)), and Gemini DR produced the most citations per report in DeepResearch Bench without leading on accuracy ([DeepResearch Bench](#)). Third, **verification by a human skim fails as well**: unassisted PhD experts reached only **60.8% accuracy** when verifying claims in deep-research reports ([DeepFact summary](#)) **M** abstract-level. A report has to make each load-bearing claim cheap to check, with the quote and the link sitting next to it.

Practitioners reached the same diagnosis without the numbers. Benedict Evans showed OpenAI's own demo citing Statcounter and Statista and misreporting Japan's smartphone split. He argued that a tool that is "mostly right but not reliably right" forces you to check everything, which cancels the time saved ([Evans](#)) **P**. FutureSearch's six-query test found the same pattern of SEO and company-blog sources, outdated figures and misattribution, including citing Statista after the agent had already found the original PDF ([FutureSearch](#)) **P**. The most upvoted skeptical position on HN was "the error rate is unknown and hard to detect", not "it is useless" ([HN 43133207](#)). Simon Willison's 2025 work traces the arc. By September he judged GPT-5 Thinking's in-reasoning search better than the dedicated Deep Research mode for most queries ([Research Goblin](#)). A month later he called a \$1.10 o4-mini-deep-research run "unverified slop" after checking 1 of its 19 results, and noted that tool-call limits make deep research unsuited to exhaustive lists ([Willison TIL](#)) **P**. That last point matters for engineering research. A Webhound tester found only ~3% of known-available data after 20 minutes ([HN 45373008](#)) **P**. "Find *all* X" tasks belong to code that queries structured APIs, not to an LLM browsing loop.

Engineering research adds two risks of its own. Staleness is the first: deprecated-API use is measured, and versions, prices and maintenance status change monthly. The second is **default bias in tool choice**. An independent study of 2,430 Claude Code responses found that it most often *builds custom code instead of recommending a library* (the most common label in 12 of 20 categories). When it does name a tool it commits hard (GitHub Actions 93.8%, Stripe 91.4%), and its defaults skew toward a JS stack (Vercel, Drizzle, Tailwind) that reflects model preference rather than market adoption ([Amplifying](#)) **M** small JS-heavy sample. For build-vs-buy research, the prompt therefore has to force the agent to list existing options before it proposes a build.

Effort, retrieval and independent verification move the numbers; wording mostly doesn't

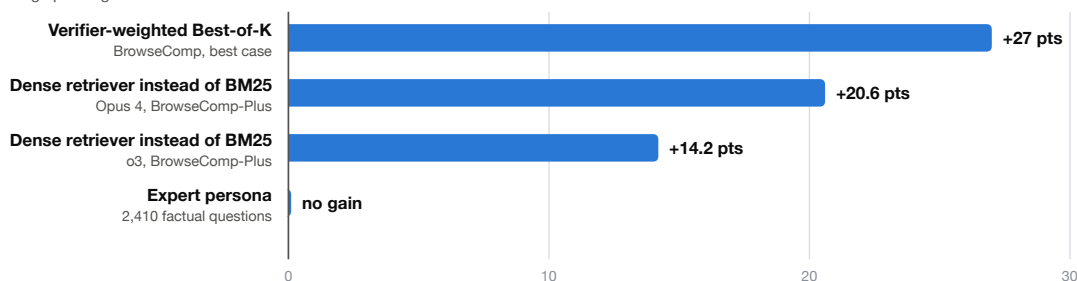
The interventions with measured effect act on the system, not the phrasing. In Anthropic's analysis, **token usage alone explained 80% of BrowseComp performance variance**, and token use, tool calls and model choice together explained 95%. A multi-agent setup (Opus 4 lead, Sonnet 4 subagents) beat single-agent Opus 4 by **90.2%** on an internal eval, at roughly **15x** the tokens of chat ([Anthropic engineering](#)) **V** not independently replicated. Retrieval quality is equally first-order. Swapping BM25 for a dense retriever raised o3 from 49.3% to 63.5% and Opus 4 from 15.5% to 36.1%, and search-call counts varied tenfold between models, so some models simply under-search ([BrowseComp-Plus](#)) **M**. Verification is cheaper than solving. GLM-4.5 needed ~75 tool calls to solve a task but ~18 to verify a candidate answer, and verifier-weighted Best-of-K added up to **27 absolute points** on BrowseComp, with little gain where verifying is as hard as solving ([arXiv 2510.06135](#)) **M**. Giving agents a URL-liveness checker cut non-resolving citations **6-79x, to under 1%**. It checks only that the page exists, not that it supports the claim ([Rao et al.](#)) **M**.

INTERVENTION	EFFECT	GRADE	IMPLICATION FOR THE WORKFLOW
More search effort / tokens	80% of variance (BrowseComp)	V	Set budgets by question type; don't starve hard questions
Better retriever / search API	+10 to +22 pts per model	M	Tool choice is a first-order lever; benchmark on your own queries
Verifier-weighted sampling	Up to +27 pts	M	Verify candidates in a separate pass

INTERVENTION	EFFECT	GRADE	IMPLICATION FOR THE WORKFLOW
URL-liveness tool	Dead citations to <1%	M	Give the agent a link checker; don't let the model type URLs
Multi-agent fan-out	+90.2% internal, ~15x tokens	V	Only for breadth-heavy comparisons
64-sample aggregation	+15-25% (Deep Research)	V	Rarely worth the cost for reports
Clarifying questions	Clarification quality correlates with report quality (one system)	M (correlational)	Keep it, but cheap and bounded
Expert persona	No gain over control on 2,410 factual questions; not reliably better on GPQA/MMLU-Pro	M	Drop it for accuracy
"Think step by step"	Marginal for reasoning models at 20-80% extra time	M	Drop it

Measured accuracy gains from system changes vs a prompt-wording change

absolute percentage points gained



Sources: arXiv 2510.06135; BrowseComp-Plus; Zheng et al. (persona: no gain over control). Also measured: a URL checker cut dead citation links to under 1%.

Figure 3. System changes move accuracy by double digits; the most common prompt-wording habit moves it by nothing measurable.

The persona and CoT results come from [Zheng et al.](#), [Wharton Report 4](#) and [Wharton Report 2](#). The clarification evidence comes from [DeepResearch-ReportEval](#), plus the argument in [Clarify-Then-Search](#) that early ambiguity propagates into wasted search. No controlled causal test of clarification exists.

Two refinements stop "add verification" from becoming slop of its own. First, **verification must use a different method or context, not more of the same model's opinion**. Steel extracted and ran Claude Code's built-in deep-research workflow: five phases, five parallel searchers, up to 15 sources with 2-5 claims each, and three skeptic verifiers per claim, where two rejections kill the claim. It criticised the majority vote because errors from the same model are correlated. It also noted there is no loop-back: one question took ~13 minutes, 98 agents, 533 tool calls and an estimated 2.36M tokens to produce 7 findings, and flagged thin evidence without chasing it ([Steel](#)) **P**. Re-fetching a page and string-matching the quote, resolving a DOI through Crossref, or reading `pushed_at` from the GitHub API are independent checks. A second LLM agreeing is not. Second, **generic "double-check your work" text in the main prompt is counterproductive on current Claude models**. Anthropic's Opus 5 guidance warns that such instructions cause over-verification and wasted tokens. It also warns that "be conservative / only high severity" makes the model under-report, so the better pattern is to ask for everything, label it, and filter later ([Prompting Claude Opus 5](#)) **V**. The verification belongs in the workflow as a structural step with tools, not as an exhortation.

The architecture that the published systems converge on is a pipeline: **clarify** → **brief** → **plan** → **search/fetch under a budget** → **notes with sources** → **lossless compression** → **verify** → **write** → **cite or check**. Anthropic splits it into a lead, subagents and a separate citation agent ([lead prompt](#)). OpenAI and

LangChain's open_deep_research put a clarifier and a brief-rewriter in front of the research model ([OpenAI cookbook](#), [ODR prompts](#)). Only Anthropic's cookbook, Jina's node-DeepResearch and ODR's legacy grader have a real critic step **P**. Multi-agent is a poor fit when subtasks share context or depend on each other, a limit Anthropic itself names for most coding work ([Anthropic engineering](#)) **V**. DeerFlow 2.x gives the cleanest rule: delegate only when the benefit (parallel latency, specialist capability, context isolation) clearly exceeds the cost (duplicate discovery, coordination), and treat a subagent's "completed" as unaccepted until it is checked as holds / does not hold / UNVERIFIED ([DeerFlow lead prompt](#)) **P**. For engineering questions the strongest check is the one Willison uses in his 2026 async research repo: have a coding agent *build the thing*. That covers testing HTTP range requests, compiling the `servo` crate, and trying smolvm as a sandbox, which turns claims into observed results ([Willison research tag](#)) **P**.

The tool layer is a first-order lever, and it churns monthly

Because retrieval quality moves accuracy by double digits, the tool configuration matters more than any single sentence in the prompt. The independent evidence on search APIs is weak and contradictory. AIMultiple's 100-query test (AI/LLM topics only, GPT-5.2 judge plus a 10% human check) ranked Brave first (Agent Score 14.89), then Firecrawl 14.58, Exa 14.39, Parallel Pro 14.21 and Tavily 13.67, but the confidence intervals overlap and only Brave versus the lowest-ranked API was significant ([AIMultiple](#)) [M-ish, single snapshot]. Parallel's own September 2026 report puts Parallel and Perplexity at the top (74 on its 50-question BrowseComp subset, versus Exa 70 and Tavily 66), and it admits vendor bias and ~2 points of noise per question ([Parallel](#)) **V**. Across both, Brave and Exa stay top-tier on raw retrieval and Tavily stays mid-to-low. The gaps among the leaders are within noise, so choose on features (filters, content extraction, ZDR, price) and on a test with your own queries.

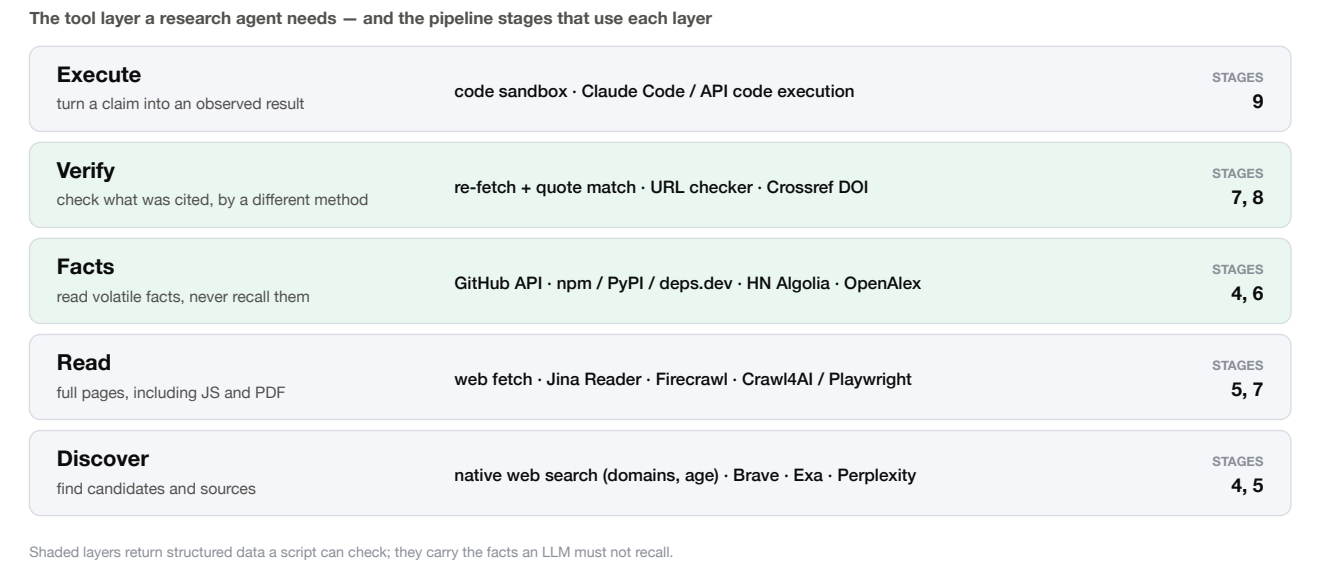


Figure 4. Five layers, each mapped to the stages of the workflow in Figure 6 that use it.

LAYER	OPTION (VERIFIED 2026-10-25)	PRICE	WHY IT MATTERS FOR ENGINEERING RESEARCH
Native search	Anthropic web_search (docs)	\$10/1k + tokens	allowed_domains / blocked_domains , max_uses , page_age , citations always on with ≤150-char cited_text ; web_search_20260209 filters results with code before they enter context

LAYER	OPTION (VERIFIED 2026-10-25)	PRICE	WHY IT MATTERS FOR ENGINEERING RESEARCH
Native search	OpenAI web search (pricing)	\$10/1k + content tokens	Built in; less parameter control documented here
Search API	Brave (page)	\$5/1k	Own index; LLM Context endpoint returns extracted text, tables and code with per-source age ; Goggles reranking; free-credit figure contradicts itself on the page
Search API	Exa (pricing)	\$4-7/1k; Contents \$1/1k pages	Fast tiers; contents and highlights in the same call; \$10 free monthly
Search API	Perplexity Search (pricing)	\$5/1k (\$1 fast)	Top in Parallel's vendor test; up to 5 queries per billed call
Search API	Tavily (credits)	1-2 credits; PAYG \$0.008/credit; 1k free/mo	Easy start; mid/low in every ranking found
Search API	You.com, Linkup, SerpAPI (You , Linkup , SerpAPI)	\$5/1k; ~\$5-6/1k; \$25 per 1k at entry	SerpAPI-style scrapers are becoming the main route to Google results (below)
Fetch	Firecrawl (pricing)	1 credit/page incl. PDF page; 1k free	#2 in AIMultiple; failed 403/404 pages still bill
Fetch	Jina Reader (page)	Token-based; 10M free tokens per key	<code>r.jina.ai/</code> prefix, renders JS, handles PDFs; ReaderLM models are CC-BY-NC if self-hosted
Fetch	Crawl4AI (docs), Playwright MCP (repo)	Free, self-hosted	86k stars / 38.6k stars, both pushed 2026-10-22; JS-heavy pages
Code facts	GitHub REST search (docs), GitHub MCP (repo)	Free	Code search 10 req/min, auth required, default branch only; the repo API gives stars, <code>pushed_at</code> , <code>archived</code> , <code>licence</code> , which are the facts an LLM must not recall
Code facts	deps.dev, HN Algolia	Free	Both returned HTTP 200 on a live probe (package metadata; practitioner threads by date)
Papers	OpenAlex (limits), Crossref, arXiv, Semantic Scholar (API)	Free	OpenAlex free key = 10x budget (\$1/day); S2 returned HTTP 429 unauthenticated; Crossref resolves DOIs for citation checks

Two deprecations change the landscape. Microsoft **retired the Bing Search APIs on 2025-08-11**, and the replacement does not expose raw results ([Microsoft](#)). Google's Custom Search JSON API is **closed to new customers, with full-web search ending by 2027-01-01**, though that comes only from secondary sources, one of them a competitor ([Brave](#)). Any harness built on those two needs a migration path. MCP servers churn in the same way. The reference `modelcontextprotocol/servers` repo has archived its Brave and GitHub servers in favour of vendor-maintained ones ([repo](#)). The vendor servers for Exa, Tavily, Brave, Firecrawl, Perplexity, Jina, GitHub and Playwright all had pushes within about a week of 2026-10-25 ([exa-mcp-server](#), [brave-search-mcp-server](#), [firecrawl-mcp-server](#)). Browserbase's MCP repo is archived ([repo](#)). The community arXiv server is active ([arxiv-mcp-server](#)). Stars are a weak signal. Last push and the archived flag are the usable ones, and a research agent should read them from the API every time it recommends a repo.

The minimum useful kit for applied research is one quality search API, or native search with domain and recency parameters, plus one full-page fetcher, the GitHub API, a package-registry API, a URL checker, and a code-execution sandbox for build-to-verify. Perplexity's own guidance supports putting hard constraints such

as domains and date windows into **request parameters rather than prose**, and taking URLs from structured search results rather than asking the model to write them ([Perplexity prompt guide](#)) . The fabricated-URL figures above show why.

Repos and products: borrow designs, verify before depending

Commercial deep-research products split into hosted reports and API primitives. Google's **Gemini Deep Research agents** (`deep-research-preview-04-2026` and `deep-research-max-preview-04-2026`) are the most complete first-party agent API found. They run on the Interactions API only, with a collaborative plan you can approve or refine, Google Search, URL Context, code execution, remote MCP servers and File Search, and are billed at standard model and tool rates. They are still labelled preview ([Gemini API docs, pricing](#))  primary. OpenAI lists `o3-deep-research` at \$10/\$40 per million input/output tokens and `o4-mini-deep-research` at \$2/\$8 ([o3 page](#), via search snippet). ChatGPT's deep research reportedly moved to GPT-5.2 in February 2026 and added site restriction and mid-run steering ([The Decoder](#)) [secondary]. Perplexity's Sonar Deep Research costs \$2/\$8 per million tokens plus search and citation fees ([Perplexity pricing](#)). Anthropic sells primitives rather than a hosted deep-research endpoint: web search, web fetch, code execution and Managed Agents. Claude Research in the apps is the orchestrator-worker system described in the engineering post ([Claude support](#)). Search vendors now sell research endpoints too (Exa's agent/research requests, You.com's Research API at \$12-1,200 per 1k calls, Linkup Research at \$0.25-2.50 per request), and their quality is untested here ([Exa](#), [You.com](#), [Linkup](#)). Elicit, Consensus, Undermind and scite target literature review, not applied engineering research. No rigorous independent 2026 head-to-head of hosted products exists. The one 2026 comparison found comes from an AI-visibility vendor with undisclosed criteria ([Presenc](#)), and DeepResearch Bench changed its judge to GPT-5.5 and froze the old board ([MLAnthology](#)). The practical consequence: build a 20-query eval set from your own past research questions before committing to any product.

Open-source repos are most valuable as **readable prompt designs**, and their stars track neither freshness nor quality. Status is from the GitHub API on 2026-10-25.

Last push to the default branch, open-source research agents

checked on the GitHub API, 2026-10-25 · stars in thousands at right
older than 6 months

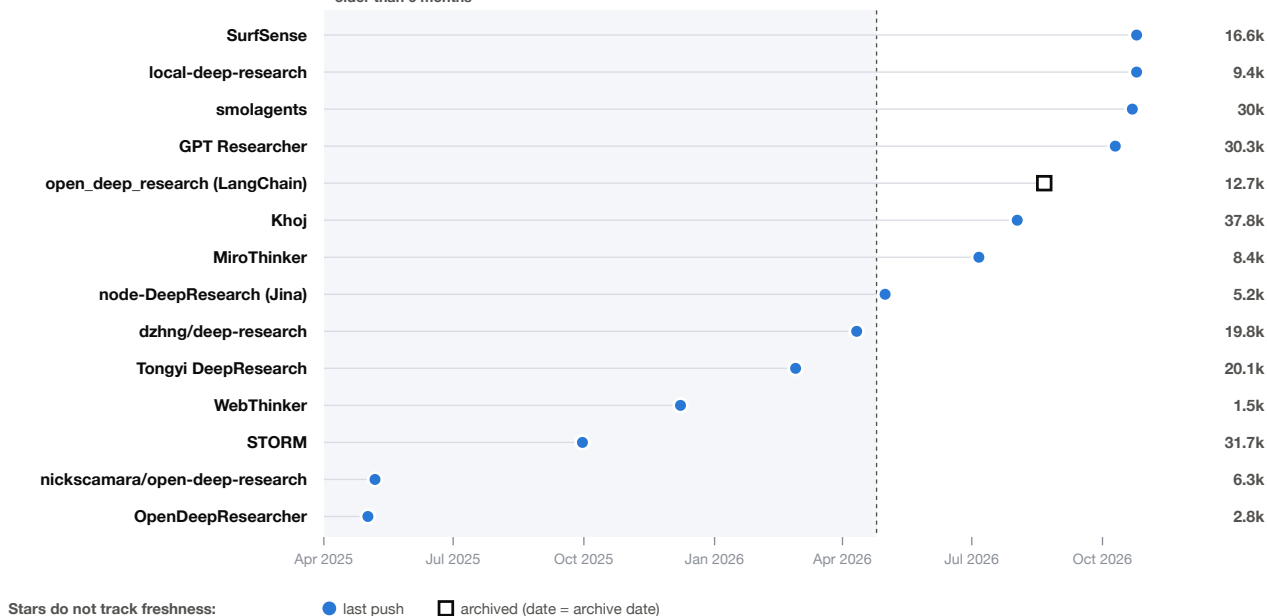


Figure 5. Six of fourteen repos have not been pushed for six months or more, and one is archived. Stars say nothing about it.

REPO	STATUS	LEARN FROM IT	AVOID COPYING
anthropics/claude-cookbooks research lead / subagent / citations	Active	Query-type classification (depth/breadth/straightforward), effort table, delegation-brief checklist, source red flags (speculation words, marketing, nameless sources), pass conflicts upward, citation pass that must reproduce the text byte-for-byte	"MINIMUM of five" tool calls; "always at least 1 subagent"; refers to a <code><writing_guidelines></code> block missing from the file; tool names tied to its harness
openai-cookbook DR notebooks	Active	Clarifier (ask only questions that change direction; don't invent preferences) plus rewriter (first person, unstated dimensions left open-ended, primary sources)	Broad rewrites of generic queries; the notebook itself warns they inflate verbosity and cost
langchain-ai/open_deep_research	Archived 2026-08-21	Clarify JSON with "almost never ask twice"; worker stop rule (≥ 3 relevant sources or last 2 searches repeat, cap 5); lossless compression with the reason stated; "bias towards single agent"	"Fairly long and verbose" sections; use as reference, not dependency
jina-ai/node-DeepResearch	Borderline stale (2026-05)	Narrow pass/fail critics (definitiveness, freshness with a max-age table by question type, plurality, completeness); orthogonal decomposition with $<20\%$ overlap (planner)	Ideological finalizer lines, "extend the content", all-caps "beast mode"
bytedance/deer-flow	Active, 84k stars	2.x net-benefit delegation test and holds/does-not-hold/UNVERIFIED acceptance; deep-research skill with temporal-precision rules	1.x reporter with 10,000-15,000-word "non-negotiable" floors and contradictory citation rules; planner that gathers more "even if 90% certain"
assafelovic/gpt-researcher	Active	Queries in plain language with no search operators (backends return empty); "take a concrete opinion" paired with cite-only-provided-sources; subtopic writer that knows what earlier sections said	1,000/2,000-word minimums, "as long as you can", persona picker

REPO	STATUS	LEARN FROM IT	AVOID COPYING
stanford-oval/storm	Stale (2025-09)	Perspective-guided questions; an expert that answers only from gathered info and says "cannot answer"; a polish pass that may only delete	Wikipedia-style output
dzhng/deep-research	Borderline stale	Tiny readable recursion; learnings keep exact numbers, entities and dates; URLs appended by code, not by the model	"The source is irrelevant"; "3 or more pages, include ALL the learnings"
huggingface/smolagents	Active	Minimal DIY base; worker may return a clarification request	Benchmark-style shortest-answer reformatter
Alibaba-NLP/DeepResearch , MiroThinker	Model repos	Only if self-hosting RL-trained research models; MiroThinker keeps only the latest K tool results in context	Benchmark claims are self-reported

The table shows that **length floors are the most common anti-pattern in published research prompts**, appearing in GPT Researcher, DeerFlow 1.x, ODR v2, dzhng and Jina. That matters because LLM judges, including any judge used to grade a prompt, reward length: length-controlled AlpacaEval raised its correlation with Chatbot Arena from 0.94 to 0.98 by regressing length out ([Dubois et al.](#))  read via summary. A floor rewards padding and fools the evaluator at the same time.

What the reusable prompt and its workflow should contain, and what to strip

The vendor guidance is consistent on form. Be explicit and give the reason behind each rule. Delimit sections with XML. Put long material first and the query last, which Anthropic reports improves answers by up to ~30% on multi-document inputs. Phrase format instructions positively ([Claude prompting best practices](#)) . Avoid contradictory or vague instructions, which make the model spend reasoning reconciling them, and give context-gathering budgets with an escape hatch ([GPT-5 prompting guide](#)) . Every serious agent prompt read injects the current date and uses numeric budgets . The structure that follows is a **stable rule block plus per-task brief slots**, re-tuned whenever the underlying model changes. Thoroughness language written for older models makes current Claude 5.x and GPT-5.x models over-search and over-verify ([Opus 5 guidance](#)) .

Include

STAGE	ELEMENT	CONCRETE CONTENT	EVIDENCE
Clarify	Bounded clarify gate	Ask only questions whose answer changes structure, depth or direction; 2-6 at most; ask about unknown acronyms; never a second round; skip when the brief already fixes target, focus and scope (DeerFlow's "2 of 3 dimensions" trigger)	 (OpenAI) +  correlational (ReportEval) +  (ODR, DeerFlow)
Brief	Rewrite into a first-person brief	Keep every stated preference; mark needed-but-unstated dimensions as open-ended; invent nothing	 (OpenAI),  (ODR)
Brief	Applied-research slots	Decision the research feeds; audience; constraints (stack, licence, hosting, budget, compliance, region); recency window; must-answer questions; candidates already known or ruled out; definitions of ambiguous terms; what "done" means (shortlist size, depth)	 (OpenAI, Anthropic brief checklist),  (Evans on undefined "adoption")

STAGE	ELEMENT	CONCRETE CONTENT	EVIDENCE
Rules	Current date plus staleness rule	Inject today's date; versions, prices, maintenance status and API shapes must be read live, never recalled; put the year in time-sensitive queries	M (deprecated APIs), P (universal in published prompts)
Rules	Effort scaling with caps and stop rules	Classify the question (straightforward / depth / breadth) and map each class to a tool-call budget; stop when the last 2 searches converge, when ≥ 3 independent sources agree, or at the cap; escape hatch: proceed and report gaps	V (Anthropic , GPT-5 guide), P (ODR)
Rules	Search tactics	Short, broad queries first, then narrow; plain language without operators; fetch full pages before relying on snippets	V (Anthropic subagent), P (GPT Researcher)
Rules	Engineering source hierarchy with reasons	Official docs, changelogs and release notes; the repo itself (code, releases, issues); package registries; vendor pricing pages (for price only); independent benchmarks; practitioner threads. Down-rank listicles, aggregators, AI-written roundups and vendors' pages about competitors; flag speculation words and marketing	M (AI-generated sources), V (Anthropic's SEO-farm failure), P (FutureSearch)
Rules	Enumerate before building	List existing OSS, commercial and managed options before proposing a custom build; "build" is one explicit option with reasons	M -small (Amplifying)
Rules	Conflicts surfaced	Report disagreeing sources with dates instead of resolving them silently; prefer primary and newer	V (Anthropic subagent)
Rules	Permission to not know	"Not found", "insufficient evidence" and "near miss" (wrong year, parent company) are valid outputs	V (OpenAI on hallucination , Anthropic , Perplexity)
Rules	Claim ledger	Every load-bearing claim gets a URL taken from tool results, a short verbatim quote and an access date; a claim without a quote is retracted or moved to Unverified	V (Anthropic quote-then-claim), M (citation-precision gaps)
Rules	Evidence-status labels, not felt confidence	Label claims verified-primary / corroborated / single-source / inference / unverified, based on what was checked	M basis (91% calibration error); scheme U, similar to DeerFlow 2.x (P)
Rules	Fact vs inference split	Keep sourced findings separate from the agent's own recommendations	U (recommended by the designs, never measured)
Tools	Hard constraints as parameters	Domains, date windows and result caps go in tool parameters; the prompt states which tools exist and what each is for	V (Perplexity , Anthropic web_search)
Tools	Structured fact sources	GitHub API for stars, last push, archived flag, licence and releases; registry/deps.dev for versions; HN Algolia for discussion; OpenAlex/Crossref when papers matter; sandbox to build-to-verify	M (retriever effect), P (Willison)
Verify	Separate pass using a different method	Re-fetch each load-bearing URL and confirm the quote appears and supports the claim; check URL liveness; re-read numbers from the primary source; re-pull repo metadata	M (verification asymmetry , URL checker), V (CitationAgent)
Verify	Narrow binary critics	Pass/fail on definitiveness, freshness against a max-age table, completeness against the brief's must-answer list, and plurality (shortlist size)	P (Jina evaluator)
Verify	One targeted second hop	Thin evidence on a load-bearing claim triggers one focused follow-up search before writing	P (Steel)
Output	Answer first	Recommendation or verdict in the opening lines; caveats short and only where they change a decision	V (Opus 5)

STAGE	ELEMENT	CONCRETE CONTENT	EVIDENCE
Output	Candidate table with defined columns	Name, link, type (OSS/commercial/managed), what it does, maturity signals with dates, licence, price with an as-of date and source, fit against constraints, risks, evidence status	V (OpenAI tables with described columns, specific figures)
Output	One citation convention used by every stage	Pick inline links <i>or</i> a numbered list, and make worker, writer and verifier agree	P (DeerFlow 1.x contradiction)
Output	Gaps / Unverified / Rejected section	What was not found, what rests on one source, and what verification killed	P (Reddit skill , Steel, STORM)
Output	Length tied to substance	"Cover the substance; no filler sections, redundant summaries or boilerplate", repeated briefly near the end of the prompt; positive style instructions; user's language; no process narration	V (Opus 5), M (verbosity bias)
Eval	Test the prompt	~20 representative past questions with known answers; rubric judge plus human spot-check; length-controlled comparison; re-run on every model change	V (Anthropic), M (length bias)

Leave out

ELEMENT	WHY	EVIDENCE
Expert persona for accuracy	No gain over control; at most one line naming audience and tone	M (Zheng , Wharton)
"Think step by step"	Marginal for reasoning models, costs 20-80% more time	M (Wharton)
Word minimums, "as long as you can", "include ALL findings", "extend the content"	Rewards padding; also inflates LLM-judge scores	U , inferred from M verbosity bias and V guidance; no A/B found
"Gather more even if 90% certain", forced minimum tool calls or subagents on every query	Contradicts stop rules; reproduces the over-investment Anthropic had to engineer out	V (Anthropic , GPT-5 guide)
Generic "double-check", "don't hallucinate", "be accurate"	Over-verification on current Claude models; never shown to reduce errors	V (Opus 5), P
"Only report high-confidence findings"	Causes under-reporting; ask for everything, label it, filter later	V (Opus 5)
Model-stated confidence percentages used as verification	Calibration is poor	M (BrowseComp)
Asking the model to write URLs	Fabricated URLs are 3-13% of citations	M (Rao et al.), V (Perplexity)
Prose for constraints a tool parameter can enforce	Parameters are enforced; prose is not	V
Same-model majority vote as the only check	Errors are correlated	P (Steel)
Multi-agent fan-out by default	~15x tokens; poor fit when subtasks share context	V
Opinionated or source-agnostic personas, ideology lines, all-caps overrides, bare "NEVER" without a reason	Push confidence and slant; vendors recommend reasoned, positive rules	P (dzhng , Jina), V (Anthropic)
Search operators in generated queries	Several backends return empty results	P (GPT Researcher)

ELEMENT	WHY	EVIDENCE
Exhaustive "find all X" via the browsing loop	Tool-call limits produce ~3% coverage; use code against structured APIs instead	P (Willison , Webhound)

A proposed workflow: twelve stages, three actors, files as the contract

The evidence above puts the gains in the system, so this section turns it into a pipeline. The assembly is this report's proposal **U**: each part carries the grade of the evidence behind it, but the whole has not been run or measured. Three principles shape it.

1. **Each step goes to the cheapest actor that does it reliably.** Code handles anything checkable: URLs, quote strings, dates, repo metadata. An LLM handles judgement: decomposition, reading, deciding whether a quote supports a claim, writing. You make the two decisions only you can make: what the research is for, and whether the result is good enough to act on.
2. **Files are the contract between stages.** Stages communicate only through files with fixed schemas. Any stage can be re-run, moved to another model or replaced by a script without touching the others. Every sentence in the report traces back to a ledger row with its quote.
3. **Verification is a separate stage and mostly code.** It runs after gathering, in a fresh context. The LLM judge answers only the one question code cannot: does this quote support this claim?

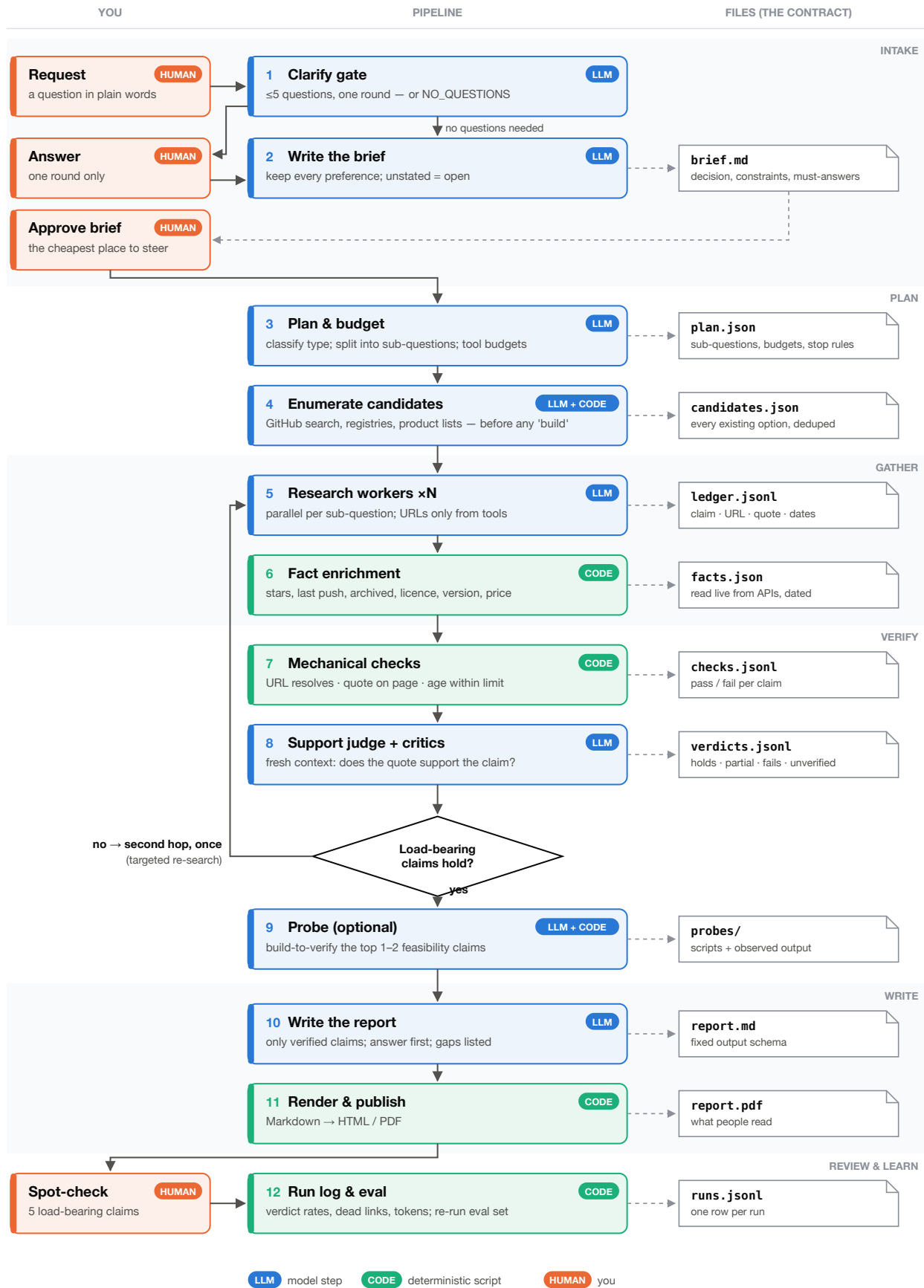


Figure 6. The proposed workflow. Blue steps are model calls, green steps are deterministic scripts, orange steps are you. Each stage writes one file, and the next stage reads only files.

#	STAGE	ACTOR	READS → WRITES	GATE OR STOP RULE	EVIDENCE
1	Clarify gate	LLM	request → questions or NO_QUESTIONS	≤5 questions, one round	V + M (correlational)
2	Brief, then your approval	LLM, you	request + answers → brief.md	you approve or edit it	V (OpenAI rewriter; Gemini's plan approval)
3	Plan & budget	LLM	brief → plan.json	tool budget per question type	V
4	Enumerate candidates	LLM + code	brief, plan → candidates.json	every existing option listed before any "build"	M (small sample)
5	Research workers xN	LLM	plan, candidates → ledger.jsonl	2 searches add nothing, 3 sources agree, or budget spent	V, P
6	Fact enrichment	code	candidates → facts.json	—	M (stale API knowledge)
7	Mechanical checks	code	ledger → checks.jsonl	—	M (URL checker)
8	Support judge + critics	LLM, fresh context	ledger, checks, brief → verdicts.jsonl	one second hop for failing load-bearing claims	M (verification asymmetry), P
9	Probe (optional)	LLM + code	top candidates → probes/	only load-bearing feasibility claims	P (Willison)
10	Write	LLM	brief, verified ledger, facts, probes → report.md	no facts outside the ledger	V
11	Render & publish	code	report.md → HTML / PDF	—	—
12	Run log & eval	code	all files → runs.jsonl	re-run the eval set on any prompt or model change	V, M (length bias)

The code stages are where the "different method" lives. Stage 6 reads stars, last push, archived flag, licence, latest version and price pages from APIs, so the report never relies on model recall for the facts that go stale fastest. Stage 7 needs only a fetch, a whitespace-normalised string match and a date comparison. That is enough to catch dead links and quotes that do not appear on the page, failures the measured 3-13% invented-URL and 5-18% dead-link rates say are common. The LLM judge in stage 8 then sees only claims that passed those checks, with the page text in hand. Its own reliability is limited: DeepTRACE's LLM judge agreed only moderately with humans ($r = 0.62$). That limit is why the review stage keeps a human.

Every claim has exactly one status; the report's sections are the end states.

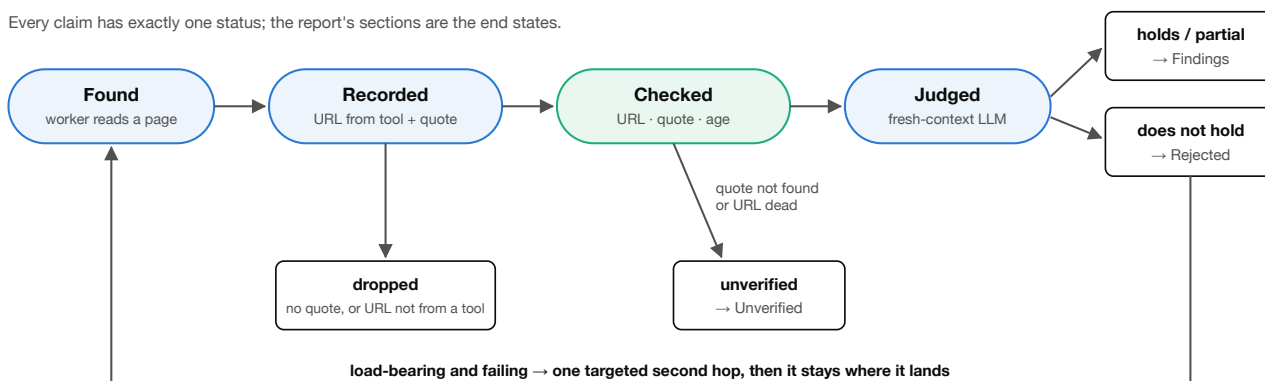


Figure 7. Every claim has exactly one status, and each end state is a section of the report. A claim whose URL a model typed is dropped, not repaired.

You touch the pipeline three times, each briefly. You answer at most five questions. You approve the brief, the cheapest point to steer, because errors propagate from the plan onward ([DeepHalluBench](#)) **M**. Gemini's Deep Research API exposes the same step as plan approval **V**. Finally you spot-check five load-bearing claims. That takes minutes because each claim comes with its quote and link. Unassisted experts verify deep-research claims at only about 61% accuracy, so the structure is what makes a quick check possible.

The pipeline scales by question type.

- **Straightforward:** stages 3-5 collapse into one agent and stages 4 and 9 are skipped. Stage 7 always runs because it costs almost nothing.
- **Breadth:** stage 5 fans out across parallel workers. This is the only case that justifies the ~15x token cost of multi-agent research **V**.
- **Depth:** one worker gets a larger budget.

In Claude Code it maps onto parts that already exist.

- A skill acts as orchestrator.
- The LLM stages (1, 2, 3, 5, 8, 10) are subagents whose prompts are in `research_prompt_v1.md`.
- The code stages (6, 7, 11, 12 and the search half of 4) are scripts: `gh api`, registry APIs, `fetch` plus string match, and `tools/build_pdf.py`.
- One run directory, `runs/<date>--<slug>/`, holds all the files.
- Workers and the judge can run on cheaper models; the planner and writer gain most from the strongest.
- Running the judge on a different model family from the workers should reduce correlated errors. This is untested **U**, motivated by Steel's critique of same-model voting **P**.

What remains untested:

- the assembly as a whole;
- its cost and running time;
- whether stage 8 adds accuracy beyond stage 7.

The eval set in stage 12 is how to find out.

Conclusion

The evidence moves the design problem out of the prompt text and into the workflow. The sentences people argue about, such as personas, "be thorough" and "don't hallucinate", are either measured as useless or untested. The parts that measurably change accuracy are the search budget, the retriever, the link checker and a verification pass using a different method, and none of them is phrasing. A reusable research prompt therefore earns its keep mainly as a *contract between stages*. It defines what the brief must specify, what counts as a sourced claim, which statuses a claim can have, when to stop, and what shape the output takes, so that tools and verifiers have something concrete to enforce. For engineering research there is a further step that report-style tools lack. Treat repo health, versions, licences and prices as data fetched from APIs at run time, and treat feasibility claims as hypotheses that a sandboxed agent can test by building. That turns "the AI said so" into an observed result, the one kind of verification that neither correlated model errors nor 61%-accurate human skims undermine.

The weakest link that remains is evaluation. No benchmark measures tool or library selection, the independent audits cover 2025 systems, and LLM judges reward length. The practical move is to keep a private set of ~20 past research questions with known-good answers and grade each prompt revision and model change on length-controlled, claim-level accuracy. Prompt advice dates as fast as the models do, and only that measurement will show which of the untested elements above are worth keeping.