

BEN DOWNEY
Co-founder and engineer. Building and measuring AI agent systems.

AN INTERVIEW ON AI EVALUATION, ENGINEERING AND EVIDENCE

Done is a claim. *I check it.* ✓

This is a sample interview with me, Ben Downey, to give you a perspective on what I've been working on: the systems I've built and how I check my work with AI. Each answer opens with the short version. Expand it for the evidence.

The work I'd take: coding and software evaluation, rubric writing, checking AI agents' work, and Brazilian Portuguese. I'd start wherever you need most. ben@bendowney.com.

Read the interview | Give it to your AI

WHAT I'VE DONE

Four things worth knowing.

25+ I co-founded a sports website platform in 2000 and still run it. It serves thousands of organizations. Nearly half its paying accounts have stayed 10 years or more.

I run a software factory of AI agents and keep refining it. It took me from about 5 to 7 finished tickets a week to about 28. The latest round checked what "done" meant. Fewer than half the jobs marked done had evidence behind them, so this round makes it prove its work.

Bot swarms kept taking the platform down. I moved over 1,000 of our domains to Cloudflare and guided hundreds of customers through moving their own DNS. Error minutes on the main site fell from about 38 a day to about 3.

I built a blog that follows our development as it happens, written and published with the same AI automation. It comes in two versions: technical field notes for engineers and plain-words guides for everyone else. It has published over 750 posts since January.

Technical timeline Plain-words timeline

HOW I WORK

Five ways I check what I build.

FIND AN ANSWER

Python, rubric, security

61 answers

01 Background and fit 15 | 02 Degree and expertise 9 | 03 Coding and systems 10 | 04 AI evaluation 12 | 05 Domain and communication 6 | 06 Working together 4 | 07 Reasoning exercises 3

EXPAND ALL

THE INTERVIEW

Answer first. Evidence on demand.

SHORT ON TIME, START HERE

Q27

My evaluation work

Q21

A real debugging walk-through

Q22

Guardrails proved by mutation tests

01

Background and fit

Twenty-five years in production. Now agent systems, and how to measure them.

Q01

What name should we use, what country or time zone do you work from, and what contact channel do you prefer?

I'm Ben Downey, based in Miami on US Eastern time. The best way to reach me is ben@bendowney.com.

Copy answer Link to this answer

Q02

Give a three-to-five sentence overview of your current professional work and relevant background.

I've co-founded and run a production web platform full stack for over 25 years, the servers, the databases and the DevOps around them, mostly JavaScript and SQL. From 2016 to 2024, before AI tools, I averaged 774 non-merge commits a year (commits that aren't just merging one branch into another). Now I build and measure AI agent systems, and most of my work runs through AI under layers of review.

DETAIL

- 6,967 non-merge commits in the main product repository, 2016 to 2024, range 644 to 929 a year. Counted 11/1/26, merges excluded.
- The 774-a-year figure above counts 2016 to 2024 only. R9's "about 9,850" and section 3's summary count a longer, different window: June 2014 (when the git record starts) to May 2025 (when AI tooling first appears in the repository).

- No 2025 commit carries an AI co-author mark. AI tooling first appears in the repository in May 2025; two July 2025 messages mention Cursor.

LIMIT

- Git can't rule out earlier chat assistance.

Copy answer | Link to this answer

Q03

Why are you interested in AI-related contract work now, and what outcome would make this worthwhile over the next 6 to 12 months?

I want to add AI contract work, and the biggest draw is working in industries I haven't worked in before. I like finding where AI helps people and then proving it does. In 6 to 12 months, success is steady work on real problems in more than one industry, with results that hold up when someone checks them and a client who wants me back.

Copy answer | Link to this answer

Q04

Rank your top three interests: AI model evaluation; benchmark/scenario design; prompt and rubric writing; coding/automation; data analysis; domain research; quality assurance; or another area.

First, coding and automation: building evaluation tools. Second, AI model evaluation and verification: checking finished work against evidence. Third, prompt and rubric writing.

Copy answer Link to this answer

Q05

What are two achievements that best demonstrate your capability? Specify the problem, your contribution, tools, and result.

First, moving a live platform behind Cloudflare. Bot swarms kept hitting us, and moving every customer's domain by hand was too big a job. I used AI at every layer, from gathering each domain's DNS data to tailoring messages by customer situation.

I moved over 1,000 managed domains myself and guided hundreds of client-managed domains through their own changes. When the next swarm pushed CPU near 100%, a block at the edge brought it back to about 20%. Second, auditing my own agent system. I asked whether a job an agent called done could be proved done. In a real week, 280 of 637 finished jobs (44%) had real evidence behind them; 8 more had a check recorded that didn't meet the rule; 349 had no check at all. That result is why I'm rebuilding how my system verifies finished work.

280 of 637 jobs had evidence. 8 more had a check recorded that didn't meet the rule; 349 had no check at all.

Copy answer Link to this answer

Q06

Which coding languages and technical stacks can you use independently? Rate each 0 to 4 and give one concrete example.

On your 0 to 4 scale: 4 in JavaScript, SQL, HTML and CSS, and production web servers and databases. 2 in Python and recent stacks, which I build with AI under my direction.

0 = none, 1 = introductory, 2 = working, 3 = advanced, 4 = expert.

JavaScript

August 2021: fixed a season rollover bug that duplicated football stat rows, by changing the server-side JavaScript that archives players and adding a 240-line staff debug page to find which rows to delete.

SQL

May 2019: wrote a 1,425-line reporting page whose queries count distinct customers, sum sales and left join several tables, to show sales by day, month and site.

HTML and CSS

April 2020: started the merged staff screen with 3,066 lines of new CSS, including 22 media-query blocks, replacing two separate screens.

Production web servers and databases

I built the production database behind the platform, including its tables, keys and indexes, and I've managed it ever since. I also wrote its original backup procedure, a date-based rotation that keeps backups closer together for recent dates and further apart for older ones.

Python

My Python is 2026 work, the measurement kit and agent tooling, written by AI agents under my direction. That's why I rate it 2.

Recent stacks, built with AI under my direction

TypeScript, React, Next.js, Node and Express, PostgreSQL. In 2026 I led the build of a multi-tenant business app with row-level database security, a tested ledger, and deploys that roll back if a health check fails. I also moved a legacy staff system to a React and Express API one slice at a time, while the old system kept running.

`LIMIT`

- These examples are in private repositories; I can walk you through them. On the recent-stacks app another person also committed, and its production status isn't checked here.

SAME EVIDENCE Q19

Copy answer | Link to this answer

Q07

Which AI assistants, development environments, APIs, or orchestration tools have you personally used? What did you actually build or evaluate?

Claude Code is my main tool. I built my agent system on it, with hooks (small scripts that run automatically at set points), gates and MCP servers (a standard way for an AI tool to reach outside data and tools) I designed and AI mostly wrote. Before that I used ChatGPT, and I've built with Cursor and Kiro.

I call the Claude API directly in a staff assistant (Q20) and use Codex and Manus as second-opinion reviewers. My system connects to Anthropic, OpenAI, Gemini, DeepSeek, Groq and OpenRouter. It spends from prepaid plans first and touches metered credits only with my approval.

DETAIL

- Cursor appears in the product repository's commit messages and rules files from July 2025 onward. Claude Code is how the agent system and the measurement kit were built.

Copy answer | Link to this answer

Q08

Have you performed AI output evaluation, test-case creation, dataset QA, red-teaming, benchmark design, or rubric scoring? Describe any examples.

Yes: evaluation, test cases and rubric scoring. I defined an audit that samples agent requests and scores how each one finished against the evidence. Its verification rules agree with 17 of 20 labeled cases.

I ran it on a real week of my own agent system (Q27). I also run multi-model review: the same work goes to several AI reviewers and I judge their findings. It isn't a formal benchmark, but it shows me where each model is strong and where it misses. I haven't done red-teaming, dataset QA or benchmark design as formal work. Their core is what I already do: define what counts as correct, check results against evidence, and record where a method fell short.

DETAIL

- The kit's verify command runs its rules against 20 labeled cases: 17 of 20 agree, 2 lenient misses, 1 strict miss. Rerun: python -m agent_scorecard verify, 11/1/26.

LIMIT

- An AI graded the study sample and wrote the 20 labels. No person has checked them yet.

WHAT I DECIDED

Defined the question the audit asks. Drafted first: drew the sample, made the first labels and wrote the code.

SAME EVIDENCE Q27

Copy answer | Link to this answer

Q09

What practical experience do you have with Python, SQL, spreadsheets, PDFs, CSVs, JSON, data pipelines, and reproducible data analysis?

SQL is where I'm strongest. I've built custom export pipelines right into my software, generating spreadsheets, PDFs, CSVs, JSON, whatever format a client actually needed. With Python I work differently: I direct AI to write it, then I read the code and check the output myself.

Copy answer Link to this answer

Q10

What domain expertise could you bring to source-intensive AI tasks?

Software engineering and SaaS operations: payments, support, releases and incidents, from over 25 years running a production platform. Amateur sports administration: registration, schedules, rosters, tournaments. A year of operating AI agents on real work.

Examples: finance, audit, trading, accounting, operations, law, engineering, healthcare, science, logistics.

And fluent Brazilian Portuguese.

Copy answer Link to this answer

Q11

What public portfolio, code samples, writing samples, GitHub profile, or open-source contributions would you be comfortable sharing?

Three small Python tools, public on GitHub under my org, DXdev. They're my own projects, built out of my agent system, not contributions to someone else's. agent-scorecard measures how much agent work finished and how much of it you can trust, run on a seeded synthetic log. agent-guardrails is a set of Claude Code hooks that stop a coding agent before a mistake I already know about. agent-memory keeps an agent's memory files within load limits, with no dead links or duplicates. My DXdev posts on verifying AI work explain the thinking behind all three.

READ MORE

DXdev on GitHub

agent-scorecard

agent-guardrails

agent-memory

Verify, do not assume

All green does not mean all tested

Verify the AI reviewer's checkable claims

A rubric verifier for autonomous workers

Copy answer Link to this answer

Q12

How many hours per week could you realistically commit, in which time zone, and when could you begin if an independently approved opportunity becomes available?

Part time works for me, something like 10 to 20 hours a week, US Eastern time, and I could start this week.

Copy answer Link to this answer

Q13

Do you have a reliable computer, internet, browser, spreadsheet/document tools, and a workable remote collaboration setup? Any limitations relevant to the work?

Yes. A dedicated multi-monitor desktop, reliable internet, Chrome, Google Docs and Sheets, and years of remote work. I also keep separate local and cloud Linux servers for isolated sandbox work.

Copy answer Link to this answer

Q14

Which work settings suit you best: structured specifications, ambiguous research, independent development, rapid iteration, written review, or team collaboration? Why?

Rapid iteration. I break work into stages, define done for each, and measure the result. Short cycles let me course-correct fast and keep everyone in sync.

Written review is valuable to me when it serves a real purpose.

Copy answer Link to this answer

Q15

What skills or work requirements would you want help with before starting? What important information have we not asked?

I'd want a walkthrough of your rubrics, quality standards and formal evaluation methods, so my work meets your bar from day one. One thing you didn't ask: how I work with AI day to day. This page is an example.

AI drafted it from my repositories and notes, a separate AI model reviewed each answer, and I made the final call on the wording.

Copy answer Link to this answer

02

Degree and expertise

What I claim, the evidence for it, and what I leave to another expert.

SME01 List each earned degree with the exact credential title, major or specialization, and institution if you wish. Clearly distinguish completed, in progress, and nondegree training.

I've got a B.S. in Computer Science, with a minor in Electrical Engineering, Magna Cum Laude, from Rensselaer Polytechnic Institute, 2003. My coursework included machine learning.

Copy answer | Link to this answer

SME02 Which academic courses, research, capstone projects, laboratory work, or specialized studies best demonstrate your knowledge? What did you personally do?

Two classes stand out. In machine learning, a homework assignment might be just a couple of sentences, but it could take dozens of hours to solve. I built prototype applications, including solutions to multi-armed bandit problems, and worked through the proofs of how the methods applied.

In another class, I programmed a remote control car to follow a line of tape using LEDs and sensors, adjusting motor speed and direction as the sensor readings changed.

Copy answer | Link to this answer

SME03 In which professional fields have you held substantial responsibility? Include position or capacity, approximate

I co-founded a sports SaaS platform and have been its engineer since 2000. I've owned payments, support escalation, releases and incidents the whole way, and I wrote most of the server and browser code myself

duration, core work, and notable outcomes.

before AI tools ever showed up. The platform serves thousands of organizations today, and nearly half of our paying accounts were opened 10 or more years ago.

Copy answer | Link to this answer

SME04

Name up to five topics in which you could credibly review, correct, or assess a specialist AI response. For each, explain why you are qualified to do so.

Five. First, JavaScript and SQL web application code: I've built and maintained a production platform in it for over 25 years. Second, whether an AI agent's finished work is really finished: I direct agents daily and check their results against tests and the real system.

Third, rubrics for agent output: I define the rules and tests my agents are measured against. Fourth, payments for a small SaaS: I've run my platform's payment side, including PayPal and Stripe. Fifth, releases and incidents on a live site, which I've handled on the same platform.

Copy answer | Link to this answer

SME05

For each proposed specialty, identify supporting evidence: education, industry practice, published writing, permitted work samples, certifications, significant projects, or public portfolio links.

The evidence for each of those sits under Q19 for the web application code, Q27 for checking whether agent work is really finished, Q28 for rubrics, and SME03 for payments, releases and incidents, plus the blog posts tied to each. I'm not claiming any certifications or published papers here.

Copy answer Link to this answer

SME06 Which adjacent fields do you understand at a working level but would NOT claim as a subject matter expert? Where would you want another expert to check your judgment?

Finance, accounting, law, healthcare, machine learning research and advanced statistics all sit right next to my work, and I'm not claiming expert level in any of them yet. I'm glad to work on them alongside a specialist, and I want to grow into them. In each one, I'd want that specialist checking my judgment on the calls that depend on field knowledge.

Copy answer Link to this answer

SME07 What type of specialist work would you most enjoy: technical coding assessment, engineering, mathematics, science, finance, business operations, industry specific writing, model evaluation, or another domain?

Building evaluation tools first, then model evaluation.

Copy answer Link to this answer

SME08 Give two potential résumé accomplishments, naming the real problem, your personal contribution, tools or methods, and defensible results.

First, I moved a live platform behind Cloudflare to stop repeated bot swarms. Using AI at every layer, I moved over 1,000 managed domains myself and guided hundreds of client-managed domains through their own

Do not invent metrics or disclose restricted work.

changes. Its protection layers and firewall API now let us react a lot faster when something hits.

Second, I built an AI agent system that multiplies my own development throughput. It runs my real development work, using Claude Code, Jira and GitHub, with hooks and required tests as the controls. My median went from about 5 to 7 resolved tickets a week up to about 28 over the last 26 weeks.

Copy answer | Link to this answer

SME09

What additional credentials or evidence could you readily provide later through official application channels? Do not put transcripts, identity documents, private employment records, or sensitive personal details in this shared document.

Education and employment verification can go through the official application channels when the time comes. I'm keeping anything sensitive out of this document.

Copy answer | Link to this answer

03 </> Coding and systems

Systems I've built, and how I debug and test them.

Q16

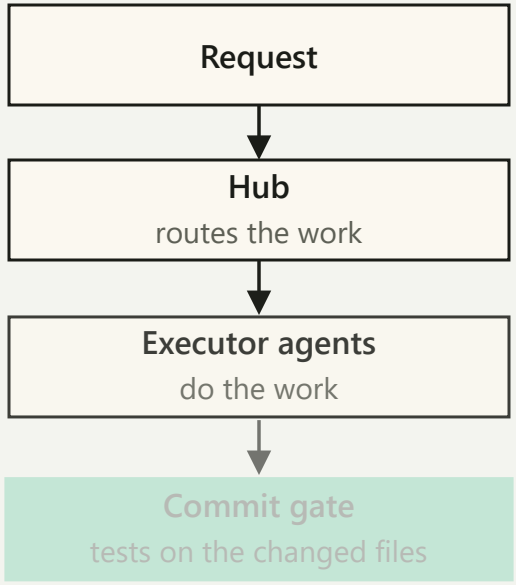
Describe one system or automation you built end to end. What inputs, outputs,

My AI agent system runs my real development work end to end. I feed it my requests and tickets, and what comes out is tested code, reviews and status reports. It depends

dependencies, controls, and measurable results did it have?

on Claude Code, Jira and GitHub, plus other AI models I bring in for second opinions.

I built in controls: hooks that block known mistakes, required tests before anything gets committed, and a rule that metered AI spending needs my approval first. My median went from about 5 to 7 resolved tickets a week to about 28 over the last 26 weeks.



The agent system the kit measures. A request goes to a hub, then to executor agents. Their work passes a commit gate that runs tests on the changed files and reaches the main branch. A spend policy and bounded loops apply at every step.

Copy answer Link to this answer

Q17

How do you use Git, branches, pull requests, code review, and collaborative version control?

I use Git Flow, with a production branch and a staging branch for weekly releases. Every developer also keeps their own branch and shares work for review on a private, staff-only site. We hold a weekly sprint release meeting where we go over what shipped and refill the task list in Jira.

In my agent system specifically, a commit gate (tests that must pass before code can be committed) also runs tests on exactly the files being committed, every time.

Copy answer Link to this answer

Q18

Describe a workflow you built or maintained that processed documents or messy structured data. How did you handle malformed, missing, and duplicate records?

Clients import their schedules straight from their own spreadsheets. The importer guesses each column from the patterns in the data, and if it can't tell, it asks the client to pick. It normalizes key data like dates and matches each team to an existing team or creates a new one.

Then a few layers of client review flag any skipped rows and say why.

Copy answer Link to this answer

Q19

Which programming tasks can you complete without an AI assistant, and which do you routinely complete with one?

Without an assistant, I can write server and browser JavaScript and SQL on my own. That's what I did for years on a production platform before AI tools ever showed up. With one, I routinely build my agent system and Python tooling, plus modern web applications in TypeScript, React, Node, Astro, Postgres and Cloudflare Workers, including migrating my platform's staff tools over to React.

Copy answer | Link to this answer

Q20

What experience do you have with LLM APIs, tool calling, structured outputs, retrieval, agents, token costs, context limits, and model version changes?

I call the Claude API directly for small production jobs, including a forced tool call with a cached prefix and schema-constrained JSON where the evidence quotes get checked in code. I've also built MCP tools for agents to use, and most of my multi-agent work runs through Claude Code. For retrieval, I use SQLite full text and TF-IDF over my own conversations and notes, not embeddings.

I track cost per call against a price table I re-verify myself, plus daily token caps. On context limits, my agents checkpoint and flag themselves at set token levels, and an idle sweep compacts background agents and resumes them. Before I move work to a new model, it goes through a price and benchmark check first.

Copy answer | Link to this answer

Q21

Walk through a real debugging problem. How did you reproduce the failure, isolate its cause, and prove the fix?

A staff member reported Staff Center pages taking 20 seconds or more to load. The monitoring data showed the same slow pattern recurring over a weekend while server load stayed low, which ruled out a traffic spike and pointed at the page itself. I traced it to two inline page sections running an unindexed OR query against a 1.64 million row table, a full scan every time.

I fixed it in stages so I could tell what moved the number. First I added the two missing indexes on the production database: load dropped from about 20 seconds to about 1.5 seconds with no code change. Then I rewrote the query to use them, moved both sections to load after the page, and confirmed the same improvement on the test environment.

DETAIL

- A ticket with monitoring data and before-and-after load times, and one commit whose verification note records the staged measurements.

LIMIT

- The numbers come from the monitoring tool and my own measurements at the time; nobody independent re-measured it afterward.

Copy answer Link to this answer

Q22

How do you build tests or validation checks for your work?

I write the check first and then try to break it. In my agent system a commit is refused unless the test gate is green. Every new guard is proved by mutation: I break the code on purpose and confirm a test goes red.

Mention unit tests, integration tests, edge cases, fixtures, reproducibility, or comparison against trusted outputs where applicable.

From the commit history: the 11/1/26 blog publishing change says 8 mutations proved red. The lane-manager state machine shipped with 33 tests and 5 mutations red. A verifier hardening change records 44 tests failing before the fix, 261 passing after, and 19 of 19 guard mutations red. Process and git code also gets separate integration tests run against real temporary git repositories.

DETAIL

- Hooks stop known mistakes at the moment an agent tries them, such as printing a credentials file into the conversation, a wait loop with no deadline, or a database query likely to scan a whole table. Four of them are public, with both blocked and allowed cases tested; 292 tests pass today.
- My measurement kit has nine test files; I reran them on 11/2/26 and 124 passed. Its fixtures are a seeded synthetic log, so every run is reproducible, plus 20 labeled cases used as trusted outputs for the verification rules. The privacy scan is tested by planting leaks. An AI wrote the 20 labels, so I had a second AI reader judge the same 20 cases without seeing any label: it agreed on 18 of 20.

SAME EVIDENCE Q16

READ MORE

agent-guardrails

agent-scorecard

Copy answer Link to this answer

Q23

How comfortable are you with spreadsheet formulas, lookups, pivots, Power Query, scripts, and workbook QA? What have you actually used?

I've used all of them and I'm comfortable with each, though spreadsheets aren't my main tool. My daily data work is SQL, where joins, grouping and aggregation are the ideas behind lookups and pivots. When a project lives in a workbook, I work in it directly and check it the way I check code.

Copy answer Link to this answer

Q24

How do you preserve source provenance, versioning, timestamps, and auditable calculation steps when merging multiple files?

When I combine data from several sources, I want anyone to be able to retrace how every number got there. I lock the date and any randomness up front, so someone else can rerun it and get the same rows. Every figure says where it came from: calculated straight from the data, rebuilt by rerunning code, typed in by a person, or missing.

A missing figure names the source it would need instead of guessing. Every percentage shows its count too, like 45 of 69, so nobody wonders what it's a share of. In my measurement kit, the input log carries a fingerprint of its contents so you can tell if it changed, timestamps without a time zone are rejected, and every version is in a dated changelog.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q25

How do you manage API secrets, client confidentiality, access permissions, local files, and AI tool uploads on a professional assignment?

Secrets live in a folder that git ignores, and a hook refuses any command that would read or print those files into the conversation, because the transcript is saved and re-sent every turn. The agent gets a value passed straight into the program that needs it, never displayed. I watched that hook fire while writing this answer, on a command that only mentioned a secret file.

DETAIL

- Client material stays out of public artifacts. My public tools run on synthetic data, and the measurement kit has a privacy scan that checks its files, file names and full git history. Two secret scanners run over that public repository found nothing.
- For AI tool uploads I have a written spending and data policy. Nothing uses a paid, per-call AI service without my explicit yes in that conversation, and background jobs never touch one. The one free-tier tool, whose provider may train on prompts, is barred from my product code, customer data and partner material. On a client job I would follow the client's own rules on which AI tools may see their files before using any of them.

READ MORE

agent-guardrails

Copy answer Link to this answer

04

AI evaluation

A good answer reads well. An evaluated answer has a check that could have failed.

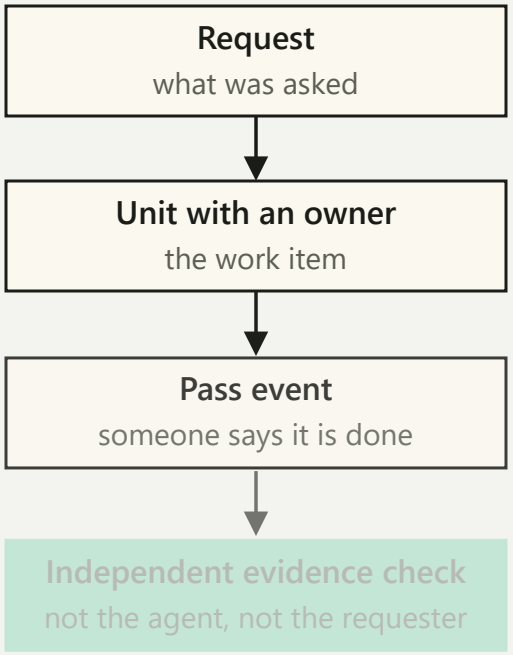
Q26

What distinguishes model evaluation or benchmarking from simply getting an AI system to produce a good answer?

A good answer reads well. An evaluated answer has a check that could have failed and did not, recorded by someone other than its author.

DETAIL

- The kit's method states it: a unit is trusted when an independent check recorded evidence of the outcome, and the verifier must not be the executor or the requester.
- My scorecard reports two numbers, and the strict one gives no credit for a finish that is only claimed.



From a request to evidence. A request becomes a unit with an owner. The unit gets a pass event. An independent evidence check records the outcome, and only then does the strict count include it.

READ MORE

Related post: verify, do not assume

Copy answer Link to this answer

Q27

Describe any evaluation you designed, performed, or audited. What was the test

The best example is one I ran on my own work. I took a real week of my agent system's log and asked a simple

population, method, success measure, and failure analysis?

question: when an agent said a job was done, could I prove it? Of 637 finished jobs, 280 (44%) had real evidence behind them, like a test that actually ran; 8 more had a check recorded that didn't meet the rule; 349 had no check at all.

The rule is a plain text match against evidence fields, no model call. That's why I'm now rebuilding how my system verifies finished work, and part of that is already live.

DETAIL

- A second one: an AI judge scores my blog posts before they publish. I had it blindly re-score 10 posts three times each. Two of its four checks agreed every time. The other two didn't, and 3 of the 10 posts flipped between pass and fail across runs. Now each post gets three runs, the majority wins, and a check that still splits is marked unresolved.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q28

How would you build a fair, objectively scoreable rubric for a complex task involving multiple source files?

I write the rubric before I open any output, so the outputs can't shape the standard. Each line checks one thing against one named source, with a short fixed set of levels, a plain definition of each, and a worked example. I always add a "cannot tell" level so a grader is never forced to guess, and if the rubric changes I re-score everything that was already scored.

That is how my measurement kit works: six labels with written definitions, and 20 test cases written before the rules ran. An AI wrote those cases, so I had a second AI reader judge them without seeing any label, and it agreed on 18 of 20.

READ MORE

A rubric verifier for autonomous workers

Copy answer Link to this answer

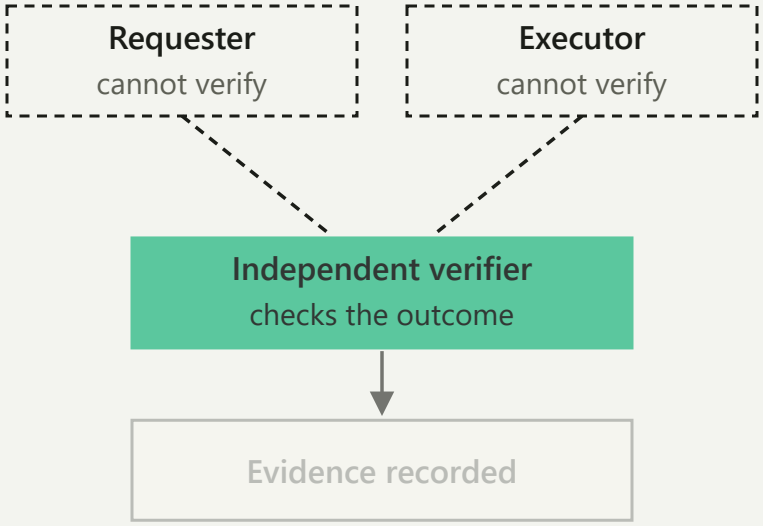
Q29

How would you prevent expected answers, reference solutions, or evaluation criteria from leaking into a model's task instructions?

The commonest leak I have seen is the agent's own report counted as evidence, and my kit excludes that by rule. A strict finish has to be backed by something other than the agent's own report.

DETAIL

- For leaks into instructions, I separate two cases. My kit's checks are plain rules with no model call anywhere, so there is no prompt for an expected answer to leak into. For a model that reviews or grades, the practice is to hand it facts to check, never the verdict I expect. The same goes for the second reader on the kit's 20 cases: it saw each case's unit, event and evidence, shuffled and renamed, and never the label.



Who may verify. The requester and the executor are both excluded from verifying. A separate verifier checks the outcome and records evidence. Only then is the unit trusted.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q30

How would you assess reliability across repeated model runs rather than treating one pass or failure as conclusive?

I never trust one run. I run the same task several times and report how many passed out of how many, with a 95 percent interval. A task that passes sometimes is unreliable.

A task that never passes is usually broken or out of reach. Those need different fixes, so I keep them apart. I also fix the seed and the as-of date, so a rerun differs only because of the model, and I have a second reader score a subset and compare their agreement with what chance would give. I learned this the hard way with my blog's AI judge: re-scoring 10 posts three times each flipped the verdict on 3 of them, so scoring now takes the majority of three runs. My measurement kit works the same way. It prints an interval on every sampled share and counts a finish as unstable if it was reopened or reverted within 14 days.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q31

How do you write realistic, unambiguous instructions without coaching the model toward the intended result?

I write the task the way a real requester would, with the actual inputs and constraints, and I say what done means and what evidence counts. What I never include is the expected answer, a hint toward it, or the grader's criteria. Then I try the prompt on a case where the obvious shortcut gives the wrong result, because a prompt that only works when the shortcut is right is coaching in disguise.

That's how I set up the second reader on my kit's 20 labeled cases. It saw each case's unit, event and evidence, shuffled and renamed, and never the label, and it agreed with the labels on 18 of 20.

Copy answer Link to this answer

Q32

How do you distinguish a minor formatting error from a meaningful factual, logical, evidence, or compliance failure?

My test is whether the conclusion would change. A formatting error doesn't change it. A wrong fact, a broken chain of evidence or a missed requirement does.

I score the two separately, so a style problem can't hide a missing outcome and a tidy answer can't hide a wrong one. The tricky cases are the ones that look cosmetic. An outside review of my kit found that a verifier name made only of spaces counted as a trusted pass. That looks like a whitespace nit, but it let an unverified finish count as verified, so I treated it as a real defect and fixed it in release 0.3.0.

SAME EVIDENCE Q28

READ MORE

agent-scorecard

Copy answer Link to this answer

Q33

How would you verify an AI-generated claim that cites a PDF section, spreadsheet cell range, or dated source record?

I open the source, find the exact page, section or cell range, and check that it says what the claim says. Then I recompute any derived number myself from the raw cells or rows. A citation I can't open isn't a citation, so the claim stays unsupported until I can.

For dated records, I also check the date against the claim's timeline. You can see the habit on this page. The 774 a year in Q02 is 6,967 non-merge commits over nine years, recounted from git. The 44% in Q27 is 280 of 637, and I redid that division too. I do this with git, tickets and live pages every day, and PDFs and workbooks get the same steps.

Copy answer Link to this answer

Q34

Describe how you would handle records with conflicting statuses, effective times, approval histories, or later corrections.

I start by freezing an as-of date. Then I rank the sources by how independent they are: the live system first, then git, then the ticket, then a transcript. I keep each event's effective time and recorded time as two separate facts.

A later correction replaces the original only when it points at what it replaces, and "cannot tell" is allowed to be the answer. I've run this on a real case (Q39): a ticket said STAGING/LIVE, but the live site still showed the old behavior, so the live site outranked the ticket until a merge and a second check showed the fix live.

READ MORE

The Hotfix That Lied

Copy answer Link to this answer

Q35

When using AI to accelerate your own work, which checks do you personally perform before trusting or releasing the output?

Before I release anything an AI helped with, I check it against the real artifact: the commit on the main line, the live page, the command output. It's a written rule in the instructions my agents work under. A change is done only when it's been tested in the real target state, and a claim that could be checked has to be checked or labeled unverified.

I do the same to the numbers on this page. Before I put the kit's test count here, I reran its tests myself, and 124 passed on 11/2/26. For anything other people read, I edit the words myself. I cut what isn't relevant and replace complex wording with plain words.

READ MORE

Verify the AI reviewer's checkable claims

Copy answer Link to this answer

Q36 How do you document uncertainty, missing evidence, or an apparent defect without inventing an answer?

I label it, and I say what would settle it. My kit's scorecard prints NO DATA, along with the source it would need, instead of a number. Goal progress and idle time are two lines like that, because the log has no source for them.

My sampling audit has a "cannot tell" label for a request where the evidence doesn't show either way, and it never counts as finished. For a defect I haven't confirmed, I write down what I saw, what I haven't checked, and the check that would confirm it.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q37 If asked to improve the difficulty of a synthetic benchmark, how would you do

I'd make it harder by adding reasoning, not ambiguity: more sources to reconcile, a dated correction that changes the answer, or a tempting shortcut that happens

it while keeping the task fair, realistic, and answerable?

to be wrong. Every item has to stay answerable from the materials it ships with, and I'd check that by having an independent solver, human or model, reach the reference answer. A zero percent pass rate more often means a broken task than a strong benchmark, so I filter by solve rate and read the zeros before I keep them.

I also keep a few items on purpose where a grader and a sensible human disagree, because they show where the grader is wrong. My kit does that: three of its 20 labeled cases are kept as known disagreements between the grader and the label, two lenient misses and one strict miss.

READ MORE

agent-scorecard

Copy answer Link to this answer

05 Domain and communication

Dated events, written deliverables, changing scope.

Q38

Rate your familiarity (0 to 4) with any relevant domains: accounting, audit, treasury, investment operations, market data, compliance, legal documentation, and business operations. Provide examples.

Business operations 3. For over 25 years I have run the operating side of a software company: payments through PayPal and Stripe, support escalation, releases and incidents. Accounting 1: the closest I come is a tested ledger in a multi-tenant business app I led the build of in 2026.

Audit 1 and compliance 1: I designed an audit that scores agent work against dated evidence, and I run written controls on secrets and spending, which is the same evidence discipline but not a financial audit or a regulatory program. Treasury, investment operations, market data and legal documentation 0. Those are fields where I'd work alongside a specialist (SME06).

Copy answer Link to this answer

Q39

Describe a time you reconciled conflicting information or made a decision that depended on the sequence of dated events.

A hotfix ticket said STAGING/LIVE, with a fix version and a real commit attached, but customers were still hitting the broken behavior. The sources disagreed, so the order of events decided it. The commit existed, but only on a hotfix branch.

The main line had no trace of it because the merge, the one step that puts a hotfix into production, had never run. I didn't trust the ticket or my own checkout. I merged, then checked the live site from outside. The first check still showed the old behavior, and a check minutes later showed the fix live. Three dated facts settled it: when the commit landed, which branch it landed on, and what the live site served before and after the merge.

SAME EVIDENCE Q34

READ MORE

The Hotfix That Lied

Copy answer Link to this answer

Q40 What types of professional deliverables can you produce to a high standard?

Scripts and tools, technical specifications, method documents, READMEs that state their limits, and scored sample tables. I've also built spreadsheet, CSV, PDF and JSON exports for clients in whatever form they needed. I write review reports that separate what I checked from what I assumed and name the source for each claim.

For example: spreadsheets, scripts, technical specifications, decision memos, source citations, and review reports.

My measurement kit is a good example of a full handoff set, and it's all public: a README, a method document, a schema specification, a glossary, a first-pilot guide and a dated changelog.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q41 How do you prefer to receive feedback, manage disagreement, and communicate a substantive quality issue?

I like feedback written, specific and tied to evidence, because then I can check it and act on it. When we disagree, I settle it by looking at the artifact. If the reviewer is right, I fix it and write it down.

If I think they're wrong, I say what I checked. When I raise a quality issue myself, I say what I saw, what it affects, how I know, and what I'd change, and I turn it into a changelog entry plus a fix. My kit has two cases where the reviewer was right. A reviewer pointed out that the README said there were no confidence intervals while the method depended on one, so I corrected the README in release 0.2.0. Another outside review found that a verifier name made of only spaces counted as trusted, and I fixed it in release 0.3.0 with a regression test.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q42 What is your process for clean handoffs, change logs, file naming, source control, and final quality checks?

Every release gets a dated changelog entry with four fixed headings: what we learned, what changed in the method, what it breaks for you, and what is still open. A test checks that the version in the package, the code and the newest changelog heading all agree. For source control I use git with a commit gate that runs the tests on exactly the files being committed, and I check the result before I call anything done.

My measurement kit shows the handoff end to end: three dated releases, a tagged first release, and a "what this breaks" note on each so the next person knows what to change.

READ MORE

agent-scorecard

Copy answer Link to this answer

Q43

Describe your experience delivering quality work to a deadline with a changing scope or incomplete requirements.

My clearest example is rebuilding our coach and staff screens into one people and roles screen, on a single ticket from April to August 2020, about four and a half months. The scope kept moving while I built it. A groups feature got folded in, a second ticket was merged into it, and design changes kept arriving as comments from the team.

I handled that in small steps the team could review, and at one point I sent a written list of what was still pending. It came to 275 commits, all mine, with roughly 89,000 lines added and 52,000 removed across source files, not counting built copies of the stylesheets. Recounted from git on 11/2/26.

Copy answer Link to this answer

06

Working together

A first assignment, and the rest is a conversation.

Q44

Rank the work you would most like to pursue immediately: evaluating outputs, writing rubrics, designing complex scenarios, building tools, validating evidence, or leading QA.

Building tools is my first pick. I've built tools for my own product my whole career, and what I enjoy most is the bridge between an interface people can use and the computation behind it that gives them results they can

act on. Validating evidence is next, because checking a finished job against git, tickets and the live system is what my kit and my agent system are built around.

Writing rubrics is third, because rubrics are how I decide what counts as done. After those come evaluating outputs, designing complex scenarios and leading QA.

Copy answer | Link to this answer

Q45

What balance of coding, analytical review, writing, and meetings would be sustainable for you?

A mix of building and review, with writing alongside both and as few meetings as the work allows. Over 10 to 20 hours a week, I'd want a good share building tools and scripts, and the rest checking outputs against evidence and writing up what I found. A short regular check-in with whoever I'm working for is plenty.

I don't need ceremony, but I do like a clear written standard to work to.

Copy answer | Link to this answer

Q46

Are you comfortable working under written confidentiality, platform policies, access restrictions, and independent quality review? Any concerns to discuss?

Yes, no concerns. That's how I already work. My secrets stay out of repositories in files git ignores, and a hook stops an agent from printing them.

My public measurement kit uses only synthetic data, and a privacy scan checks its files and full history before anything is published. I reran that scan on 11/2/26 and it came back clean, and two separate

secret scanners found nothing on 11/1/26. I welcome independent review too. The whole kit exists to check work nobody has checked yet.

LIMIT

- Q47 and Q48, written for one specific referral, aren't shown on this public page.

SAME EVIDENCE Q25

READ MORE

agent-scorecard

Copy answer | Link to this answer

Q49

What would be a strong first small assignment for you, and what one question would you ask us before proceeding?

I'd start small: take a handful of requests, label each one against dated evidence, and write four checks that can each pass or fail on their own, the same shape as the exercise in E02. That shows how I think about evidence and what I do when two sources disagree. My one question before I start: when two dated sources disagree, which does your grader treat as ground truth?

Copy answer | Link to this answer

07

Reasoning exercises

Three practice scenarios, worked with AI. I can take a fresh one live on a call.

E01

Approval timeline: A request was submitted at 14:21. One screen says the approver's access was "active" at 14:00. A separate access event shows revocation became effective at 14:03. What would you conclude provisionally, and what source checks would make your decision defensible?

Provisionally, I'd say the approval isn't supported. The 14:00 screen only tells me access was active before the revocation took effect at 14:03. It says nothing about 14:21.

A request submitted at 14:21 can only be approved at or after 14:21, 18 minutes after the revocation, so unless something changes the picture, the approver had no access when they approved. Before saying it firmly, I'd check five things. First, the approval event's own timestamp, clock and time zone, and whether it was a standing pre-approval given before 14:03. Second, whether the revocation was backdated, meaning its recorded time differs from its effective time. Third, whether that access screen is a snapshot, and as of when. Fourth, any later reinstatement or correction. Fifth, the audit log entry for the approver's action.

Copy answer Link to this answer

E02

Rubric critique: A model says "All three requests were properly approved" and cites only a present-day status screen. The assignment requires a conclusion for each request at its submission time, supported by dated evidence. Identify the main weaknesses and draft four independently testable scoring checks.

The model judged all three requests at once, from a screen that shows today's status. The task asks for one conclusion per request, at the time it was submitted, backed by dated evidence. So the weaknesses are these: it uses current status instead of status at each submission time, it bundles three requests into one verdict, it points to no dated record or event, it never

says what "properly approved" means (who may approve, and when), and it ignores later changes.

Here are four checks, each of which can pass or fail on its own. One, each request gets its own conclusion that states its submission time. Two, each conclusion cites a dated record of the approver's access at that time, not today's status. Three, the approval event's timestamp falls inside the approver's active window. Four, a request with no dated evidence is marked "cannot determine", not "approved".

Copy answer | Link to this answer

E03

Data QA: A script sorts ledger events by their displayed clock time and adds every event except those marked “rejected” to the balance. Name at least three ways its result could be wrong and explain a safer validation approach.

It can go wrong in at least five ways. Sorting by the displayed clock time ignores the effective time, time zones and daylight saving. Events with the same time have no stable order.

Consider event status, effective time, date/time zone, and the relevant cutoff if useful.

Excluding only "rejected" still counts pending, voided, cancelled and duplicate events. Reversals and corrections get counted as new events instead of being tied to the entries they fix. And with no stated cutoff, events after the reporting date can slip in. To validate it more safely, I'd sort by effective timestamp in UTC, with an event sequence number to break ties. I'd accept only a named list of final statuses, remove duplicates by event id, link each reversal or correction to the entry it fixes, and state whether the cutoff is inclusive. Then I'd reconcile the result to a trusted opening and closing balance and write one test case for each failure above.

Copy answer Link to this answer

WORKING TOGETHER

A first 30 days, and what I'd ask you.

What the next 30 days would look like, and what I'd want to settle with you first.

R10

First 30 days

In week one I'd read your existing graders' transcripts and the labeled set, so I know where your standard sits. In week two I'd run a reliability pass on one evaluation, checking whether two graders reach the same verdict on the same work. In weeks three and four I'd build one hard but fair task set and have a second reader check it.

I'd shape all of this around what you need most.

Copy answer Link to this answer

R11

Open questions and sharing check

Three things I'd like to learn from you. How do you check today that two reviewers agree, and what level of agreement counts as good enough? What does a finished, accepted task look like on your side, and who signs it off?

And where do your current graders disagree most often? I'd also like to hear how you'd like me to share or use this page as I talk to other referral partners.

Copy answer Link to this answer

CASE STUDY



How we built this page.

AI and I built this page together. Each of us reviewed the other's work, and we reached a result neither of us would have reached alone.

01 Security and orientation scan.

Before drafting anything, we scanned the questions for privacy and security risks. We settled what could appear on a public page and what stayed out. Then we got oriented on what each question was asking.

02 First draft.

AI drafted every answer from my real work: my repositories, blog posts and notes. It also tried several ways to present them. Every number was traced back to its source.

03 The interview, one question at a time.

We went through the answers one by one, and most went around several times. A second Claude model read every answer as a skeptical hiring manager and sent 37 back for improvement. It also caught that my three interest rankings disagreed, so I reranked them with building tools first. I read the drafts on my phone and sent my feedback back in.

04 Presentation.

Then we worked the layers: the deliverables, styling, formatting, color and motion. The page started out red, but red reads as "fail" on a page about checking work, so we switched to green, the color of a check passing. We

tried different word orders and swapped section titles until the page felt right.

05 Tests and the proof table.

We reran the measurement kit's tests before quoting them, and 124 passed on 11/2/26. Every answer got three reviews: mine and two independent Claude model passes. The table below shows each answer and where its three reviews stand.

68 answers get this review in all. All 68 are in the JSON and markdown files; 61 show on this page.

QUESTION		ME	SECOND CLAUDE	CLAUDE
Q01	Name, location, contact	✓	✓	✓
Q02	Professional overview	✓	✓	✓
Q03	Why AI contract work	✓	✓	✓
Q04	Top three interests	✓	✓	✓
Q05	Two achievements	✓	✓	✓
Q06	Languages and stacks	✓	✓	✓
Q07	AI tools used	✓	✓	✓
Q08	AI evaluation experience	✓	✓	✓
Q09	Data handling	✓	✓	✓

Q10	Domain expertise	✓	✓	✓
Q11	Public portfolio	✓	✓	✓

06 A self-check before sending.

Before sending this page, we ran it through an AI the way a reader's AI would: once skeptical, once enthusiastic, asking how well I fit their goals. It pointed out real gaps: work I wanted wasn't named, one number had no source, two questions were missing, and some internal labels and jargon had slipped through. We fixed presentation and clarity only. No facts, numbers or answers were invented or inflated, and anything that would change the substance of an answer went back to me first. That check ran 11/2/26.

RELATED RESOURCES

Code and writing you can open.

CODE, PUBLIC ON GITHUB

DXdev on GitHub

WRITING, ON THE DXDEV BLOG

Verify, do not assume

github.com/dxdev-inc

agent-scorecard

github.com/dxdev-inc/agent-scorecard

agent-guardrails

github.com/dxdev-inc/agent-guardrails

agent-memory

github.com/dxdev-inc/agent-memory

dxdev.com/blog/agent-verification-before-claiming-success

All green does not mean all tested

dxdev.com/everyday-ai/all-green-doesnt-mean-all-tested

Verify the AI reviewer's checkable claims

dxdev.com/blog/verify-the-ai-reviewers-checkable-claims

A rubric verifier for autonomous workers

dxdev.com/blog/rubric-verifier-autonomous-workers

The Hotfix That Lied

dxdev.com/blog/jira-staging-live-never-merged-to-master

FOR YOUR AI

Give this to your AI.

The same interview as clean, stable files. Every answer keeps its id across all of them, so an agent can extract, cite and compare.

JSON

`/interview.json`

Structured answers, rated skills, review dates. Validates against the schema.

Markdown

`/interview.md`

The whole interview as one file, ids in every heading. Paste it into any chat.

llms.txt

`/llms.txt`

The standard pointer file that tells an AI agent where the clean formats are.

Schema

`/interview.schema.json`

JSON Schema for the fields, so an agent can validate before it compares.

Copy all as markdown | **Download: Markdown** | **Download: PDF** | **</> curl (JSON)**

Ben Downey. Updated 11/2/26. The same interview as JSON, markdown, PDF and llms.txt. See how every answer was reviewed.