INSIGHTS · AI WORKFLOW DESIGN · PART 1 OF 2

My AI Builds Polished, Accurate Decks. That's Why It Takes Hours.

I built a custom Claude plugin that makes C-level-ready decks, then measured every turn, tool call and token it takes. The hours turned out to be the price of the polish and accuracy I designed for.

Dan Martz, EvoNova Advisors | Insight | November 2026 | Part 1 of 2 | 12 min read

BY THE NUMBERS

11 sec

to render, check and image a finished deck in one pass. Building the fastest version from a cold start takes roughly 107 minutes.

Plugin run telemetry, November 2026

95%

of the variation in run time across 65 measured agent runs is explained by two things: output tokens written and turns taken.

Plugin run telemetry, 65 runs, 839 agent-minutes

363

turns used by today's draft stages. Thirty minutes of stages has room for about 150.

Plugin run telemetry; budget from the fitted time law

Eleven seconds to render and check a deck, nearly two hours to build one

I really like what my custom Claude plugin produces. Give it a brief and it returns the kind of deck I would put in front of a chief executive or a board: a clear storyline, action titles, every figure sourced, the pages clean. The downside is the clock. A run takes hours. And I kept finding that I needed something less polished and much quicker, for a weekly status update, say. That need is what started this analysis.

So I measured where the time goes, using the plugin's own run records: every turn, every tool call and every token. One number stopped me. Once a deck is finished, a single pass to render it, run every automated check on it and image the pages takes about 11 seconds. Yet the fastest version of the plugin needs roughly 107 minutes of run time from a cold start, and the client-ready version about 204 minutes, not counting research. The software is fast. Something else is using up nearly two hours.

The question I set myself was simple. Could a version finish in thirty minutes?

The plugin is polished, accurate and slow for the same reason: I designed it for perfection, and perfection is paid in turns and tokens. Turns are the round trips between the AI and its files; tokens are the words it writes, its own thinking included. A thirty-minute version would be doing a different job from the full run. Part 2 covers what I will change, and which kinds of confidence a quicker job can do without.

© 2026 EvoNova Advisors

Every review in the plugin found something on its first pass

First, what the hours buy. The plugin works the way a good consulting team does, in stages. It researches the question and writes a storyline. A fresh-eyes "partner", a reviewer that never saw the drafting, critiques that storyline before a single slide exists. Then the plugin builds the deck and runs quality assurance, with a devil's advocate on the research and a stress test on the finished claims. Every figure carries its source and a confidence rating. On the reference brief I test it with, research alone built an evidence register of 90 entries in about 30 minutes. Where a number is missing, the plugin leaves a visible bracket instead of guessing. Researchers argued in 2025 that language models hallucinate partly because the way they are trained and tested rewards a confident guess over an honest "I don't know".

Much of that rigor comes from cycles. A cycle is a loop: a reviewer sends the work back, the stage that wrote it revises it, and the reviewer reads it again. The storyline goes to partner review and back. The finished deck goes to QA and back, through a fix cycle.

The loops find real problems. On each of the last four quality runs, the first partner review sent the storyline back for revision, with 18 to 23 requests on the runs where they were counted, and the first full QA found between two and seven major issues. On the two most recent runs, none of those review requests was structural: the storyline's shape held, and the fixes were inside it. The research devil's advocate weakened three of the five claims it tested, and the stress test weakens five to seven claims per deck.

Every loop adds work. The client-ready run on the current version took 19 agent launches. The storyline was written three times and reviewed three times, and the build and QA each ran twice. It graded 96 on the plugin's own 100-point rubric. That grade is my rubric, applied to one reference brief: a measure of how rigorous the process is, and short of a market-tested accuracy rate.

I built for "right" before "fast" because that is what my readers worry about. In a McKinsey survey published in March 2026, 74% of about 500 organizations surveyed called inaccuracy a highly relevant AI risk. In a University of Melbourne global study of more than 48,000 people in 47 countries, fielded from late 2024 to early 2025, about two-thirds of the employees who answered its questions on using AI at work said they rely on AI output without checking its accuracy, by their own account. And in 2025 a 237-page report written for a government department in Australia was found to cite research that does not exist and to carry a fabricated quote attributed to a court. The revised report disclosed that generative AI had been used, and the consulting firm that wrote it agreed to refund part of its fee.

From a cold start, even the fastest version takes roughly 107 minutes, and cutting stages won't fix it

A weekly status update needs a quick turnaround, so the fastest version is the one to time. A word on clocks first, because AI run times are easy to misread. The plugin counts time three ways. Agent-minutes add up every agent's time, so two agents working at once count twice. Pipeline time runs from the first agent's start to the last one's finish. Then there is the clock you feel, from sending the request to receiving the file, and none of the plugin's published figures is that one. The client-ready run shows how the first two relate. It logged about 220 agent-minutes inside 204 minutes of pipeline time, and the two sit close together because its agents mostly worked one after another.

© 2026 EvoNova Advisors

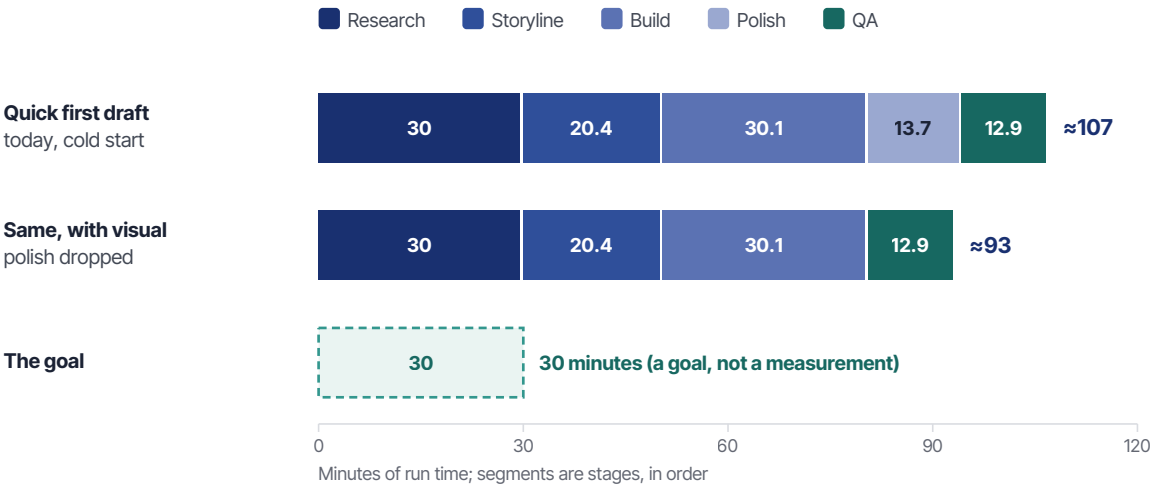
The fastest version is the quick first draft. It already skips the partner review, the de-AI pass, the stress test and the fix cycles. From a cold start it still comes to roughly 107 minutes of run time. About 30 of those go to research. The rest is four stages: the storyline (about 20 minutes) turns the research into an argument, the build (about 30) turns the argument into slides, visual polish (about 14) cleans up the look of the rendered pages, and QA (about 13) checks the finished deck. They average about 19 minutes each, and the time is spread across all four, so there is no obvious one to cut.

I assembled the 107 from two measurements taken on different versions of the plugin: research in September, on an earlier version, and the four stages on 1 November. It leaves out the intake and the hand-offs between stages, so the felt clock runs longer, and for a weekly status update the felt clock is the one that decides whether the deck is ready in time. Getting from there to thirty minutes means cutting more than two-thirds of the time.

The obvious fix is to drop another stage. Exhibit 1 shows what that buys. Take out visual polish as well and the cold run comes to about 93 minutes, or about 63 not counting research, and the deck loses the only pass that looks at the rendered pages. Research on its own takes about as long as the whole goal. The time is inside the stages.

EXHIBIT 1

Dropping stages takes a cold first draft from about 107 minutes to about 93, not thirty.



Source: author's analysis of the plugin's run telemetry, 1 November 2026. Stages are first attempts measured on 1 November under the client-ready profile, so they slightly overstate a draft's work; research (about 30 minutes) was measured on 23 September 2026 on an earlier version. The 107 is assembled from those two measurements. The draft already skips partner review, de-AI, the stress test and fix cycles.

A minute is made of tokens and turns

I took 65 measured agent runs from the plugin's own telemetry: 839 agent-minutes, about 4,000 turns and 4.4 million output tokens in all. Then I looked for what predicted each run's length. Two things were enough. The first is turns. A turn is one round trip: the AI asks for something (a file, a search, a check), gets it back and decides what to do next. The second is output tokens, the small fragments of text the AI writes, its thinking included.

By the fit, each turn costs about 3.6 seconds and each 1,000 output tokens about 8 seconds. Those two rates are the whole model, a time law that explains 95% of the variation in run time across the 65 runs. A model can flatter the data it was fitted on, so I tested it on runs it had not seen. Fitted only on runs of an earlier version, the law put today's four draft stages at 77.5 minutes; they took 77.1. Stage by stage it can miss by a few minutes, so the claim holds for the whole draft path. Its scope is narrow: one family of models, one setup, and time spent waiting on the web left out.

If you assume AI is slow because the software is slow, this is where the data disagrees. Once a deck exists, rendering it takes under a second and the full battery of automated checks about seven; with a PDF and the page images, the whole pass comes to about 11 seconds. Across a writing stage, running tools takes 1–7% of the time. Thinking is 39% of the storyline stage and 43% of the build, and it sits inside the token count, because thinking is one of the things the AI writes. Nearly everything else is round trips and words. Exhibit 2 breaks a run's minutes down and tags each part with what it buys.

EXHIBIT 2

Tokens written and turns taken explain 95% of the variation in run time across 65 runs; one pass to render and check a finished deck takes about 11 seconds.



Source: author's analysis of the plugin's run telemetry, 65 agent runs, 1 November 2026; fit on one model family in one setup, web waits excluded. The fit has no third term, so tool time sits inside the turns. One client-ready deck: 1,072 turns, 1,184 tool calls (684 of them reads), about 1.19 million output tokens. Out of sample: fitted only on runs of an earlier version, the law put today's four draft stages at 77.5 minutes; they took 77.1.

Much of the spend buys accuracy, and most of that is reading. In the four stages a draft keeps, reads make up 57–68% of the tool calls. In the storyline stage, 66 of its 99 tool calls were reads or searches, and 51 of those 66

went into the plugin's own reference files, its rulebooks on structure, evidence and style. Each of those reads is a round trip, and each round trip is on the bill.

Then come the cycles. Every fresh reviewer starts by re-reading what it is about to judge, and every send-back makes the writer re-read and rewrite. That re-reading is where both the assurance and the hours come from, and it is why a loop that finds problems on every first pass also adds turns on every first pass.

The rest buys polish, as the plugin makes it today. The build effectively writes every slide twice. The storyline stage's outline of the slides runs to about 74 kilobytes; the build then re-expresses the same slides as a deck specification of about 206 kilobytes. On one complex slide, a driver tree, 34 of its 37 elements are placed by hand, coordinate by coordinate. All of that is written out, so all of it lands in the token term. A visual-polish pass then goes over the result, and in a client-ready run a fix cycle goes over it again. Much of polish is mechanical, and mechanical work can be quick. Here the expense sits in the way this plugin produces it.

Add up a single client-ready deck and the scale is plain. Across its 19 launches, the agents took 1,072 turns, made 1,184 tool calls (684 of them reads) and wrote about 1.19 million output tokens. Run those through the law and the turns come to roughly an hour of agent time and the words to about 160 agent-minutes: about 224 between them, against the 220 the run logged. That is time summed across agents rather than on the wall clock, and this run is one of those the law was fitted on, so treat it as an illustration of the law at work. Each step is quick, and there are more than a thousand of them.

I made it inefficient on purpose

A lot of those turns are there because I put them there. I designed the plugin to read its reference files a section at a time rather than whole, to reduce hallucination, and I knew it would take many more turns. Each read stays small, and the number of reads multiplies. Before writing a word, the storyline and build stages each read about half a megabyte, one section at a time.

Picture a very fast librarian who walks into the stacks, reads one paragraph, walks back to the desk, writes one sentence, then walks back for the next paragraph. It looks inefficient. I would call it cautious. It will not carry ten books back to the desk and trust its memory of them. In the storyline stage, 51 of its 66 walks went to the plugin's own rulebooks. Each walk is quick. A thousand walks take an hour, which is roughly what the 1,072 turns of one client-ready deck come to.

The caution has a reason behind it. A peer-reviewed study published in 2024, on models from 2023, found that the models it tested used information in the middle of a long input markedly worse than information at the start or the end. Those models are dated, and I would not state the effect as a rule for today's. The study still matters because it describes the failure I was designing against. A 2025 study of 18 models, by a company that sells retrieval software, found that reliability fell as inputs grew longer. The plugin's other accuracy controls cost turns the same way. A reviewer that never saw the drafting has to read its way in before it can judge anything, and that independence is the reason to have it.

The bet has a serious counterweight. A 2024 study found that, given enough resources, letting a model read everything beat feeding it selected passages, on average. Nothing in my own data isolates what the section-at-a-time rule is worth: I have not run the same brief both ways and compared the results. It is a reasoned bet whose price shows up in the turn count.

© 2026 EvoNova Advisors

Grounding a model in sources also reduces hallucination without ending it. Commercial legal research tools, as tested in 2024, still gave incorrect or misgrounded answers 17–33% of the time. Even my own test suite still flags "no invented statistics" as a weak spot, which is why the plugin checks so much and leaves a bracket rather than a guess.

Thirty minutes means changing the shape of the work

If the minutes are the price of polish and accuracy, a faster version has to choose what it stops buying. None of the easy answers gets there on its own.

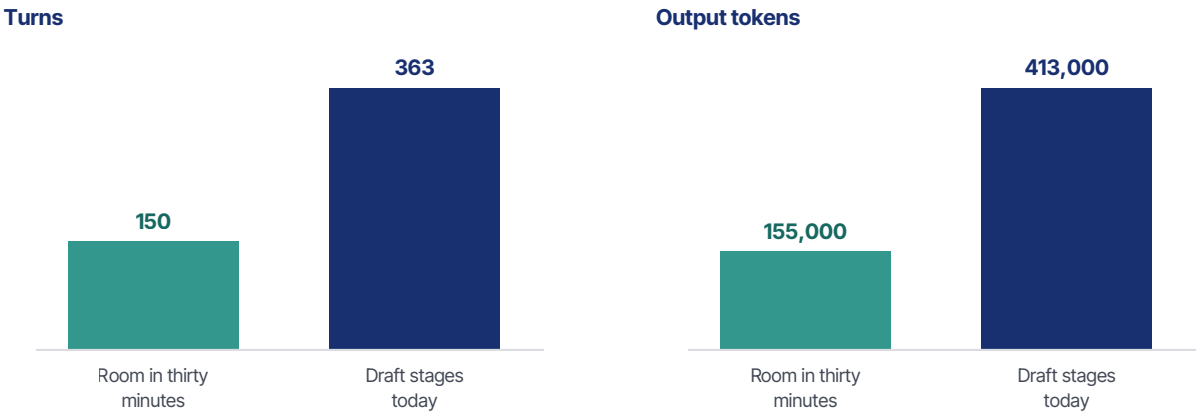
Fewer stages gets to about an hour not counting research, about 93 minutes from a cold start. Faster software has little to give, since the software's own work takes seconds a pass. A faster model is less of an answer than it sounds: in the one clean test I have, on the visual-polish stage, the quickest run came from the bigger model thinking less, ahead of the faster model. More agents help where the work divides, and much of building a storyline is tightly sequential, each step leaning on the one before. One recent preprint found that multi-agent setups helped on work that splits cleanly and hurt on a tightly sequential planning task.

Exhibit 3 sizes the gap in the units that cost time. If all thirty minutes went to the stages, run one after another, there would be room for roughly 150 turns and 155,000 output tokens. Today's draft stages use 363 turns and 413,000 tokens, and a cold run has to fit research in as well, so the real gap is wider. By the law, today's 413,000 tokens alone would take about 56 minutes, nearly twice the goal, before a single round trip is counted. So the stages have to read and write less than half of what they do now, or run side by side. In practice it will take both.

Part 2 will test four levers: reading from a prepared pack instead of walking to the stacks, writing each slide once, working side by side only where the work divides, and thinking hard only where judgment is needed.

EXHIBIT 3

Thirty minutes of stages has room for about 150 turns and 155,000 tokens; today's draft stages use 363 and 413,000.



Source: author's analysis of the plugin's run telemetry, 1 November 2026; room computed from the fitted time law, assuming all thirty minutes go to the stages run one after another. A cold run must also fit research, so the real gap is wider. Draft-stage totals are measured (412,915 tokens shown as 413,000).

A weekly status update does not need everything a board deck needs. The thirty-minute version I am after is a deck matched to that job. Part 2 is about which confidence it can do without, while keeping the craft, which today costs what it does because of how the plugin makes it.

Four questions to ask of any AI workflow

The same questions apply to any AI workflow you are building or buying. In Gartner's 2025 survey of 183 finance leaders, 59% said they use AI in their finance function, about where it stood a year earlier. Adoption had levelled off in 2025, so for many finance leaders the live question is how these workflows are designed.

1. Which clock are you measuring? Ask for the time from your request to a usable file, the one you feel, rather than a vendor's run time. My own 107 minutes leaves out the intake and the hand-offs, and I built the thing.
2. Where do the minutes go? Count round trips and words written before you blame the software, or pay to upgrade it. In my plugin, the software's pass took seconds and the words took hours.
3. Which of your accuracy controls carry the weight, and what does each one cost in time? Every review loop in my plugin found something, and every one is paid for in re-reading. A review step you cannot price, you cannot trade.
4. Before you ask for faster, which confidence are you willing to give up, and who checks the output if the workflow no longer does? My answer for a weekly status update is not my answer for a board deck.

The hours are a price I chose; a weekly update shouldn't have to pay all of it

Back to where I started: about 11 seconds to render and check a finished deck, nearly two hours of run time to build the fastest version from a cold start, and more than a thousand round trips for a client-ready one. I did not stumble into that gap. I chose it, in pursuit of perfect polish and no hallucinations, and I paid for it in turns and tokens. For a C-level-ready deck, that perfection is what I want.

The measurements turned a vague complaint, that a run takes hours, into a bill I can read: so many round trips, so many words, and what each one buys. A weekly status update is a different job. In Part 2 I will look at what has to change inside the stages to get a version to thirty minutes for jobs like that one, and which confidence it gives up to get there. Any AI workflow spends the same currency, and it is worth knowing what yours is buying.

---

evonova ADVISORS

An EvoNova Advisors Insight, Part 1 of 2. Run figures come from the author's analysis of his custom Claude plugin's own telemetry: 65 measured agent runs and timings taken on 1 November 2026, with research measured on 23 September 2026 on an earlier version. The 107-minute cold run is assembled from those two measurements. Outside studies are cited from their publishers, with their dates and limits; preprints and vendor-published research are labeled.

© 2026 EvoNova Advisors

SOURCES

EvoNova Advisors — The Thirty-Minute Deck: an analysis of the EvoNova Consulting plugin's run telemetry (65 agent runs, 839 agent-minutes; the author's own measurements) — November 1, 2026

Kalai, Nachum, Vempala and Zhang — Why Language Models Hallucinate (arXiv:2509.04664, preprint) — September 2025

McKinsey — State of AI trust in 2026: Shifting to the agentic era (2026 AI Trust Maturity Survey, about 500 organizations) — March 2026

University of Melbourne (Gillespie, Lockey) — Trust, attitudes and use of artificial intelligence: A global study 2025 (more than 48,000 people in 47 countries; self-reported) — 2025

Fortune — Reporting on the revised report for Australia's Department of Employment and Workplace Relations — October 7, 2025

Liu, Lin, Hewitt, Paranjape, Bevilacqua, Petroni and Liang — Lost in the Middle: How Language Models Use Long Contexts (*Transactions of the ACL*, vol. 12; models tested in 2023) — 2024

Hong, Troynikov and Huber — Context Rot: How Increasing Input Tokens Impacts LLM Performance (18 models; technical report published by a retrieval-software vendor) — 2025

Li, Li, Zhang, Mei and Bendersky — Retrieval Augmented Generation or Long-Context LLMs? A Comprehensive Study and Hybrid Approach (EMNLP 2024, Industry Track) — 2024

Magesh, Surani, Dahl, Suzgun, Manning and Ho — Hallucination-Free? Assessing the Reliability of Leading AI Legal Research Tools (*Journal of Empirical Legal Studies* 22(2); tools as tested in 2024) — 2025

Kim et al. — Towards a Science of Scaling Agent Systems (arXiv: 2512.08296, preprint; 180 agent-system configurations) — December 2025

Gartner — Gartner Survey Shows Finance AI Adoption Remains Steady in 2025 (183 CFOs and senior finance leaders) — November 18, 2025

An EvoNova Advisors Insight, built on the author's own run telemetry and on published research on hallucination, long inputs and multi-agent systems. Survey findings are cited with their sample sizes; preprints and vendor-published research are labeled; no figure in this piece is a price.

© 2026 EvoNova Advisors