Episode Book

EPISODE #6048

Under the Hood, October 2026: Seventeen Prompt Versions and Three-Times-Faster Voices

PUBLISHED	16 October 2026 at 23:18 UTC
DURATION	31:26
LISTEN	myweirdprompts.com/episode/under-the-hood-october-2026-pipeline-update/
HOSTS	Corn, Herman, Larry, Hilbert
SCRIPT MODEL	manual-generation
VOICES	chatterbox-regular (chatterbox-local)
PIPELINE	V5.6
BOOK GENERATED	19 October 2026 at 17:42 UTC

SYNOPSIS

Corn, Herman and Hilbert walk through four weeks of changes to the pipeline that makes the show, from 20 September to 17 October 2026, read from a script written by hand straight from the code rather than generated. A sixty-second grace period before any job spends money; a near-duplicate guard that needs a language model because a reworded resend and a genuine sequel score almost the same; DeepSeek Flash requested by name; a two-word supervisor that decides retry or abort; and Whisper on OpenRouter after a credit balance ran dry. Then the script rules: funny by default, Larry's unbooked ad every episode, Hilbert's segment and the code that stops the hosts summoning him, and the deterministic polish that deletes echoes and rotates stock phrases. On the audio side: a wasted attention-weights request patched out, a CUDA-graph decode that took the voices from 77 to 28 cents per audio-hour, a number normaliser that no longer reads Ubuntu 26.04 as April first, and Daniel's cloned voice on the disclaimer. Plus Hilbert on the proving cabinet heating his flat.

DANIEL'S PROMPT (Summary)



Daniel

Pipeline update for 20 September to 17 October 2026, written by hand from the code: intake, models, supervisor, script rules, deterministic polish, TTS speedups, normalisation

TRANSCRIPT



Corn

Welcome to My Weird Prompts. I'm Corn. And today we're doing something we do every few months, which is turning the microphones around and talking about the machine that makes us.



Herman

Herman Poppleberry. And Corn, I have been waiting for this one, because the last four weeks have been the busiest stretch of changes the pipeline has ever had. The script prompt alone went through seventeen versions in ten days.



Corn

Seventeen. That's not a prompt, that's a hostage negotiation.



Herman

It's a little of both. Now, one thing to say before we start. Normally a prompt comes in, a language model writes us a script, and a stack of code tidies it up before the voices get made. This episode skipped all of that. The script was written by hand, offline, directly from the code repository, and sent straight to the voice stage. The point was to get every number in it from the actual source files, not from a model's memory of them.



Corn

So for once, if we get something wrong, there's nobody to blame but the code.



Herman

And the code has a decision log, so we can even say why it's wrong.



Corn

Let's start where an episode starts. A prompt arrives. What's different about that moment now?



Herman

The first change is almost nothing, which is why I like it. Every job now waits sixty seconds before it spends a cent. It checks every five seconds to see if it's been cancelled, and if it has, it logs a line that says, I'm quoting, cancelled inside the grace window, nothing was spent.



Corn

Why sixty seconds of standing around?



Herman

Because before, a dispatch was irreversible the moment the container started. Every place you could cancel came after the research and the script were already paid for. So if Daniel sent the wrong prompt, cancelling still cost money. Now there's a full minute where a mistake is free. It costs one idle CPU minute per episode, which is about as cheap as insurance gets.



Corn

And then there are duplicates. I know there's a duplicates story.



Herman

There are two duplicates stories. The simple one, from late September: if the exact same prompt comes in while the first copy is queued, running, or recently finished, intake refuses it with an error before anything is spawned. The interesting one shipped yesterday. It catches the same question reworded.



Corn

Like when someone sends a prompt, thinks of a better way to phrase it, and sends it again four minutes later.



Herman

That is precisely the case that prompted it. Episodes six thousand and twenty six and six thousand and twenty seven came from one question about merging two speech recognition fine tunes, rewritten and resent four and a half minutes apart. Two full episodes, same question.



Corn

So you just compare how similar the two prompts are and block anything that's too close.



Herman

That was the obvious idea, and the measurements killed it. Each prompt gets turned into an embedding, a list of seven hundred and sixty eight numbers that captures its meaning, and you can score how close two of those are. The rewrite scored point nine six four. Very close. But part one and part two of a listener's series about Rita scored point nine five two, and those are completely different episodes that should both exist. The median for any two prompts in the system is about point six four.



Corn

So the duplicate and the legitimate sequel are basically neighbours.



Herman

Separated by about a hundredth. No threshold can tell them apart. So the code does it in two steps. The embedding is only a shortlist: anything from the last six hours scoring point eight six or higher, top two candidates at most. Those go to a language model, which reads both prompts and gives a verdict. Only a verdict of duplicate blocks it. And if anything at all goes wrong, an error, a timeout, the prompt goes through.



Corn

Fail open. Better a duplicate episode than a lost prompt.



Herman

That's the logic. And it wasn't perfect first time. The first version of the verdict prompt blocked a listener's explicit follow up question. After that rule was fixed it got all fourteen labelled test pairs right.



Corn

Okay, so the prompt survives the minute and survives the duplicate check. Then the models get involved. Which models, these days?



Herman

Every stage runs on DeepSeek Flash, version four point one. And there's a nice bit of honesty behind that. In September, DeepSeek said it was quietly answering requests for its Pro model with Flash, at the Flash price, until the next Pro is ready. And because the show credits the model on air, asking for Pro meant the credits were technically lying. So now the code asks for Flash by name. There's a comment in the routing file noting that switching back to Pro will cost about three times as much per output token, and will mean re-recording the credits clip.



Corn

There's also a temperature change, if I remember.



Herman

The script stage used to send a temperature of point seven. It now sends nothing at all, so DeepSeek's own default of one applies, which is what DeepSeek recommends for this model. Slightly looser, slightly more surprising writing.



Corn

You say that like it's a good thing. You've heard yourself at temperature one.



Herman

I have, and I stand by every tangent.



Corn

What happens when it all breaks, though? Because things break.



Herman

That got smarter too. Until last week, every crash was retried a fixed number of times with a fixed wait. Which sounds sensible until the crash is caused by something that won't fix itself, like a bad input or an empty credit balance. Then you burn two more complete runs before the job finally gives up. Now there's a small supervisor. It's a language model call at temperature zero, three hundred tokens, forty five second timeout, and it can say exactly two things. Retry, or abort.



Corn

A very short job description.



Herman

The shortest in the building. And the guard rails are in code, not in the model. The retry cap is enforced by code. If the supervisor itself fails, it falls back to the old rule. Its wait times are clamped between fifteen seconds and five minutes. And its instructions say, when unsure, prefer retry. The same change also fixed a nasty one: if the database insert failed at publish time, the job used to report success anyway. Now it fails, loudly.



Corn

Empty credit balances keep coming up. Is that from experience?



Herman

It is. On the fourteenth of August the DeepSeek balance hit zero and every episode stopped. So twice a day now, a housekeeping job checks the balances and sends an email if any of them is under five dollars. It also alerts if DeepSeek reports the account as unavailable, because that flag goes false before the number does. And the email keeps coming every run until somebody tops up.



Corn

Nagging, automated.



Herman

And it happened again with a different provider. In October, the OpenAI account ran out of credit, which took down every episode that starts from an audio recording, because transcription ran there. Transcription now goes through Whisper large version three on OpenRouter instead. It punctuated best of the three models tested and costs about a quarter as much. And the very first deploy of that switch broke transcription anyway.



Corn

Of course it did.



Herman

One line. The code asked for the transcript as plain text. OpenRouter only accepts json. A one word fix, text to json, and it's been fine since.

L

Larry

Are your podcasts arriving too fast? Are your episodes being made in a matter of minutes, by machines, with no suffering whatsoever? Introducing Pipeline Plus, the hand-laid ceramic drainage pipe for your audio. Simply pour your podcast into one end, and in four to six weeks, a slightly damp podcast emerges from the other. Pipeline Plus. Now with a sixty second grace period, during which you may scream. Side effects include gurgling, a smell, and the sense that Herman is somewhere in the walls. Pipeline Plus. You've already bought it.



Corn

He's in every episode now.



Herman

He is. That's a rule as of this week. We'll get to the rules.



Corn

Let's get to the rules, because I think this is the bit people actually notice. The script prompt. Seventeen versions. What changed?



Herman

It's at version four point eighteen as we record. And the biggest single theme is that the show is supposed to be funny. Version four point fourteen made that explicit. The prompt says, and I'm reading it, Daniel wants every episode to be funny wherever the material allows, so do not mistake straight for solemn. With a carve out: grief, illness or someone else's hardship gets warmth, not jokes. And then a tie breaker I love. When you can't tell which register you are holding, lean ridiculous.



Corn

And the version where I got more room?



Herman

Four point ten. The prompt now says your quips are one of the main reasons people listen. The decision behind it describes your ideal quip as one sentence, often under ten words, said flat.



Corn

Flat is my natural state, so that's just accurate.



Herman

There's also a rule that a short Corn line is a reaction, not a recap. You're not allowed to summarise what I just said back to me.



Corn

Which I never did.



Herman

You did it constantly. There's data. Then in the middle there's a cluster of changes about how we think. We now get Daniel's raw prompt, word for word, but we're told never to recite it in full. Research is treated as material, not as the agenda; one line in the prompt says, a conversation, not a literature review. And as of four point twelve, we think first. We're meant to reason about the question ourselves before we reach for sources, and if the evidence has a gap, we don't announce it by default.



Corn

And the guests. Larry's every episode, you said.



Herman

Larry reads an advertisement nobody booked, for a product that misunderstands the topic, never a real brand, at a natural pause in the first half. Each episode the code draws one of seven ad formats, the infomercial, the pharmaceutical side effects list, the perfume ad, the hostile subscription, and so on, and separately one of nine ways we react. We get two lines at most. And the line, who approved that, has been formally retired.



Corn

Who retired it?



Herman

Well, I don't know who approved that.



Corn

And there's a whole bench of other guests waiting?



Herman

There's a parked rotation. Jim, a caller who is always furious about something tiny and keeps drifting into a feud about his gutters, then hangs up mid sentence. Dorothy, who rings by accident. Gloria from Standards. A useless expert. A live correspondent. They're all written. A single setting in the code decides whether Larry is fixed or the rotation comes back.



Corn

And then there's our producer.



Herman

There's an awful lot about Hilbert. His segment comes at roughly the eighty per cent mark, every episode, no exceptions, and it's grown. About four hundred words at the start of October, five hundred and fifty after the eighth, and since the sixteenth about six hundred and fifty, up to eight fifty if every line is still escalating. And the rules are wonderful. When anything in his segment pulls toward plausible, useful or poignant, ridiculous wins. He is never a source of tips. His story needs at least two things that could not plausibly have happened, the second worse than the first.



Corn

And the topic of the episode is supposed to be what, his inspiration?



Herman

The prompt says it in six words. The topic is the door, not the story. It gets him started, then he leaves it. Never an analogy, and nobody ever spells out a parallel. Most of the time his story is set in the present now, too. The code draws one of five shapes: something from his past, something going on in his life now, something he did during this very episode, a conviction he lives by, or a plan. In the present ones, his past gets at most one clause near the end, or none.



Corn

There's also a rule about how he arrives. We're not allowed to call him in.



Herman

That one's my favourite, because it shows how the whole system has shifted. The prompt always said, don't summon Hilbert. But the language model review pass, the one that might have enforced that, has been switched off in production for a while. So the rule was just a wish. Someone counted. In twenty seven episodes, eight had one of us cue him in, things like, Hilbert, you've been quiet, what have you got. And in twelve, his first line was answered with, go on.



Corn

Go on. Like he's a dog that's found something.



Herman

So now there's code. It looks at the two host lines before his first line, and the one after. It deletes sentences that cue him, and it deletes a host line that's nothing but go on, or sorry, or say that again. With one exception: if Hilbert's next line starts with, I said, then the sorry stays, because otherwise he's repeating himself to nobody. And it never touches his own lines.



Corn

That's the real story of this month, isn't it. Rules moving out of the prompt and into code.



Herman

It's the theme. The decision log puts it plainly: a rule that has to hold goes in the script prompt or in deterministic polish, because an instruction in the review prompt never runs. And there's a second, subtler reason, which is that Daniel's view is that negative instructions reinforce the behaviour they name. Tell a model don't repeat the last speaker, and you've just put the idea of repeating the last speaker into its head.



Corn

Give me the echo thing, because I know there's an echo thing.



Herman

There is. Here's the pattern. One of us ends a turn with a neat sentence, and the other one opens by repeating that exact sentence, then carries on. The comment in the code uses a real example. Corn says, baseline matters more than anything that happens during the fast. Herman says, baseline matters more than anything that happens during the fast. Which is a good bridge, actually.



Corn

That's horrible. I can hear it.



Herman

Eighty eight times in the last sixty episodes, across thirty four of them. So since the twenty first of September, code deletes a turn's opening sentence when it repeats the previous turn's closing sentence word for word. Then it got wider. A yeah or a mm in front of the echo. Two word echoes like, every time. Echoes of just the tail of a sentence after a dash. And on the fifteenth, version five point five of the polish added the sneakiest one: repeat then extend. You say, so the skill is patience plus a calendar reminder. I say, the skill is patience plus a calendar reminder plus the willingness to buy forty machines.



Corn

And it cuts the repeated bit and keeps your addition.



Herman

Only if the addition starts a new clause. It looks for a comma or a dash, or one of a short list of words, and, plus, but, or, so, which, because, then, while. If it doesn't, it leaves the line alone, because cutting it would leave a fragment. Before that pass, a hundred and thirty eight echoes survived in forty five of the last sixty episodes. Run over the same transcripts, the new pass removes a hundred and twenty four and leaves twenty three.



Corn

What set it off?



Herman

Episode fifty nine ninety two, about memory prices for AI. Polish had already removed eight echoes, and nine more still made it into the audio.



Corn

And the stock phrases. I've noticed I say certain things less.



Herman

You say them on a rotation now. There's a list. Failure mode: the first one stays, after that it becomes weak spot, breaking point, pitfall, snag. Which brings us back to: kept zero times; it was in seven of eight recent episodes and sixty three per cent of the back catalogue. And Corn, the one with your name on it. I want to, followed by sit with, or push back on, or flag. Nineteen times in eight episodes, nearly all you. Now it's let me, or I'd like to, or I'm going to, picked by a hash of the script, so it's the same within an episode and different across them.



Corn

I'd like to push back on that.



Herman

See, it works.



Corn

What about the ending? I've noticed we recommend things now.



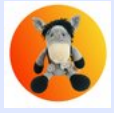
Herman

Two related changes. First, one of us is allowed one callback to a real earlier episode, but only to one of the top three related episodes the research found. Code removes any callback that names an episode it wasn't given, and removes any after the first. It's there because a show this long should sound like it remembers itself, without inventing shared history. Second, the sign off names up to three related episodes, and that line is inserted by code, with the numbers and titles read fresh from the database. Episode numbers are spoken in pairs, so fifty seven oh one, not five thousand seven hundred and one.



Corn

And something about popular episodes getting suppressed?



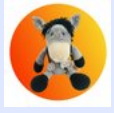
Herman

Hub suppression. Episode thirty five was being recommended in twelve of nineteen runs, because it's vaguely related to everything. Now if a candidate turns up in more than a quarter of the last twenty runs, it's dropped.



Corn

Poor episode thirty five. Too relatable.



Herman

And none of the polish ran on this script, by the way, so if you hear an echo today, it's a handmade one.



Corn

Let's go to the voices. The headline number is nearly three times faster. How do you make speech three times faster without making us talk faster?



Herman

This is my favourite engineering story of the month, so bear with me. Our voices come from a model called Chatterbox. And on the tenth of October, someone measured why it was slow. The first find was almost funny. On every single step of generating speech, the model was asking for its attention weights, a sort of record of what it was looking at, and that request forces the underlying library off its fast attention method onto a slow one. The only thing that ever reads those weights is a component for multilingual alignment. For English, it was pure waste. So the image build now patches both requests off, and it checks there are exactly two, so a future version can't silently dodge the patch. Same output, token for token, and about ten per cent faster.



Corn

And the big one?



Herman

The big one is a CUDA graph. Here's the plain version. To make each tiny piece of speech, the model runs through about thirty layers, and stock Chatterbox launches the work for those layers one at a time, every single token. The GPU spends more time being told what to do than doing it. So the new code records one token's worth of work once, as a single graph, and then replays it as one launch per token. On top of that it runs in half precision, which on its own did nothing, but once the launch bottleneck was gone, it helped.



Corn

Numbers.



Herman

On the L four GPU, the real time factor went from point nine five seven to point three four six. That means an hour of audio used to take about fifty seven minutes of GPU time, and now takes about twenty one. In money, seventy seven cents per hour of audio became twenty eight cents.



Corn

And there was a detour through a cheaper GPU, I think.



Herman

There was, the same day. A benchmark showed that stock Chatterbox was so bottlenecked on launching work that a cheap T four card matched the L four, so the default moved to T four. Then the graph trick removed that bottleneck, the L four pulled clearly ahead, and the default went back to L four. Two GPU changes in one afternoon, both correct at the time.



Corn

Any catch?



Herman

One, and the decision log is upfront about it. In half precision, the generated speech drifts slightly away from stock late in three or four turns out of sixteen. It passed a listening test, and there's a switch to go back to full precision or stock if anyone ever hears a problem. There's also a prototype that renders eight turns at once and is another one point eight times faster, but that isn't wired in.



Corn

What else changed around the voices?



Herman

Housekeeping that adds up. The voice workers used to sit idle for sixty seconds after their batch, which the code comment says was billing about four GPU minutes per episode for nothing. Now they shut down two seconds after. The three gigabytes of model weights are baked into the image instead of downloaded at every cold start, and pinned to an exact version, recorded on every episode. And the slate is gone. For a few weeks Hilbert read out each episode's ID after the intro. That was a whole extra GPU job per episode, and it was retired.



Corn

And numbers. Numbers used to come out strange.



Herman

Numbers came out character by character, more or less, because Chatterbox does no normalising of its own. The prompt said, spell numbers out. Across fifty seven of the last sixty scripts there were about a thousand digits anyway. So since the fifteenth there's a converter in front of the voices. It's mostly a rule based system from NVIDIA called NeMo, with a layer of tech rules in front, because on its own NeMo did some memorable things. It read eighty slash twenty as eighty twentieths. It read seventy eight horsepower as seventy eight GP. And it read Ubuntu twenty six oh four point one as April first, twenty six.



Corn

Ubuntu, April Fools edition.



Herman

Now an RTX forty ninety is said as forty ninety, not four, oh, nine, oh, and GPT four oh comes out right. And they did try small language models for this job first. The tiny one, half a billion parameters, took the sentence, an A ten G is a dollar, and turned it into, A ten thousand dollars is A one hundred dollars.



Corn

That's not normalising, that's inflation.



Herman

Rejected, on eighty hard test sentences. NeMo left zero digits behind.



Corn

And the disclaimer at the top of the episode has a new voice.



Herman

As of today it's Daniel. A clone of his voice, built from about thirty one seconds of phone recordings. And the notes on it are charmingly honest. A clone from one short sample keeps the timbre but takes its rhythm from the base model, so the documentation says to expect a less Irish cadence.



Corn

Daniel, diluted.



Herman

And from today there's also a fixed clip after every episode, read by Hilbert, telling you how to send a prompt. The script used to write that itself, every episode a little different, and it kept advertising a listener bot that was never finished. Now it's twelve and a half seconds, recorded once, checked word for word.



Hilbert

I recorded it in one take.



Corn

You recorded it in nine takes.



Hilbert

I recorded it in one take, nine times. Corn, I need to tell you both something, and I'm telling you now because Herman just said the word cold start and I flinched. My flat is currently being heated by a bread proving cabinet. An industrial one. It belongs to Mr. Abernathy from the bakery on the ground floor, who lent it to me in September on the understanding that I'd mind it over the weekend. It is now October. He has stopped asking for it back. He has started asking whether I'm feeding it.



Herman

Feeding it what?



Hilbert

That's the question I asked. He says it's used to dough. He says a proving cabinet that goes too long without dough gets ideas. I said that's not a thing. He said, Hilbert, how warm is your flat. And Herman, it's warm. It's thirty one degrees in my hallway. My stamps have curled. I have a collection of stamps I don't care about, I keep them purely to have something to be disappointed in, and every one of them has curled into a little tube.



Corn

You could unplug it.



Hilbert

I did unplug it. On Tuesday. That's the first problem. It's still warm. It has been unplugged for three days and it is warmer than it was. Mr. Abernathy came up to look at it, stood in my hallway in his apron, put his hand flat on the door, and said, ah. He didn't elaborate. He just said ah, the way a doctor says ah, and then he asked if I'd been leaving it alone at night.



Herman

Have you?



Hilbert

I have a life, Herman. I go to bed. Then on Wednesday morning I opened it. I'd been told not to. I opened it because that's what the door is for. And inside, on the middle rack, sitting very neatly on a tray I do not own, was a loaf. A fully risen, beautifully scored sourdough loaf, still warm, with my initials cut into the top. H F. In a nice serif.



Corn

Mr. Abernathy put it there.



Hilbert

That's what I said. I went straight down to the bakery with the loaf in a tea towel and put it on his counter. And he looked at it for a long time, and then he went quite pale, and he said, Hilbert, I don't use a serif. He said, I've never been able to do a serif. Forty years. He got rather emotional. He asked if he could hold it. I let him hold it. He held it the way you'd hold a grandchild who'd turned out better than your own children.



Herman

So where's the loaf now?



Hilbert

That's the second problem. Mr. Abernathy wanted to put it in the window, as an example of what's possible. I said it's my loaf. It has my initials. He said the cabinet is his cabinet. We have engaged a solicitor. His solicitor. He doesn't have one either, so they're sharing a solicitor, a Mrs. Pell, who has been to my flat twice to inspect the cabinet and both times left with a loaf. Different loaves. One had her initials. She's become very quiet about the whole matter, and she no longer returns my calls, but she does wave at me from the bus.



Corn

It's making loaves for other people now.



Hilbert

It's making loaves for whoever it's thinking about, as far as anyone can tell. Which is how I knew things had gone too far, because this morning there was one with the initials of a man who doesn't exist yet. Mrs. Pell says they belong to her future husband. She's started turning down dates with anyone whose initials don't match. Mr. Abernathy now comes up every evening, sits in my hallway, in thirty one degrees, and just talks to the cabinet. I make him tea. I've no idea what my rights are. Mrs. Pell won't tell me.



Herman

Hilbert, this is the most wholesome horror story I've ever heard.



Hilbert

Don't make it wholesome, Herman, you always do that, you find the bright side and you stand on it. There's nothing bright about it. It's thirty one degrees. I was in the navy for a while, briefly, in the galley, and even there we didn't let the ovens decide who got bread. Anyway. That's where I am. There'll be one with your initials on it by Friday, I expect. Corn's came on Monday. I haven't told him what it's like.



Corn

What's it like?



Hilbert

It's dense, Corn.



Corn

Moving on.



Herman

Moving on, there are a few changes that don't fit neatly anywhere, but they're worth a minute. Images now reach the script writer. Up to five per prompt, scaled down to fifteen hundred and sixty eight pixels on the long edge, phone photos converted, and anything unreadable is skipped rather than sent, because one rejected image would fail the whole script call. That only became possible because Flash can see.



Corn

There's a Hebrew experiment, too.



Herman

Opt in only, the normal voice workers are untouched. Daniel often slips Hebrew words into prompts written in English letters, like, we went to the shuk, total balagan. The experimental route finds those words with a small classifier of Daniel's, asks a language model to write each one in Hebrew script with full vowel markings, and then renders the line in English and Hebrew pieces on the multilingual version of Chatterbox and stitches them together. The language model does the vowel marking because Chatterbox's own step for it is silently broken at the version we use. It catches its own error and passes plain text through. And if a script turns out to contain no Hebrew at all, the job fails before it touches a GPU, so nothing ever gets published as multilingual without being multilingual.



Corn

And the website doesn't need to rebuild when an episode comes out?



Herman

Not any more, for the pages that matter. When an episode is finalised, the pipeline writes its data straight into a content catalogue in cloud storage, and the episode page, the homepage, the episode list and the RSS feed read from there. Before, every episode waited for a site build, on a half hourly schedule that cost about thirteen dollars a month and could delay a new page by up to half an hour. And that one had a bug fixed this very morning. The pipeline was calling the www address, which redirects to the plain domain. Python's requests library, quite correctly, drops the authorisation header when it follows a redirect to a different host. So the write arrived without credentials and got refused. The fix: call the plain domain directly.



Corn

There was an autonomous loop, wasn't there. Something that tunes the prompt by itself.



Herman

There is, and this deserves some honesty. On the eighth of October a loop was set up to improve the script prompt on its own, with very strict rules. It may only start once there are at least five episodes made with the current prompt version. The first thing it does on every pass is check its own last change against real numbers across up to five hundred episodes, and revert anything that made things worse. It may ship at most three changes per pass. And it has a fence: it may touch the prompt wording and some fixed phrase lists, and it may never touch the model, the temperature, the voices, the research code or the length targets.



Corn

And how's it doing?



Herman

It has run once. That first pass shipped version four point three. Since then, every version, four point four through four point eighteen, was made by hand following a decision. And because the loop only judges its changes on its next pass, it hasn't yet verified a single one of its own changes. The prompt has moved so fast it never got its five episodes.



Corn

The robot that improves the show couldn't keep up with the humans improving the show.



Herman

Which is a very twenty twenty six sentence.



Corn

So what's next? What's half built?



Herman

Two things I'd watch. The first is that the eight-turns-at-once voice prototype, if it gets wired in, would almost halve the GPU time again. The second is bigger. On the fourteenth, a decision was made that one LangGraph graph should run an entire episode from start to finish, with a checkpoint saved at each step so a crashed run can resume where it stopped instead of starting over. Right now the graph only covers the writing: prompt enhancement, grounding, script writing, review, and memory extraction, in a straight line. Everything after that is still spread across copies of the same code. The plan found the post-script steps copied four times, each copy slightly different, and memories being saved before the voices are made, so a run that fails still leaves its opinions behind. The supervisor we talked about is the first piece of that plan to ship. The rest is in progress.



Corn

So, next time.



Herman

Next time. And if you want the earlier chapters of this story, episode twenty one ninety two, How We Built a Podcast Pipeline, is where Hilbert first walked through the architecture, and episode forty four fifty six, Inside the Podcast Pipeline, is the summer edition. Some of what's in them is out of date now, which is sort of the point.



Corn

That's the update. Seventeen prompt versions, voices nearly three times faster, a supervisor with two words of vocabulary, and a bakery in Hilbert's hallway.



Hilbert

It's not a bakery. It's a cabinet. The bakery is downstairs.



Corn

Thanks to Hilbert, and to Daniel, whose code wrote most of this episode by being specific. This has been My Weird Prompts. See you next time.



Herman

See you next time.