

/erica-layer

AI ON PURPOSE

A PLAYBOOK

The Personalized Podcast Playbook

A five step path from "I have a long document I keep meaning to read" to "it arrives in my podcast app on its own." It is written for the case where the content is already good and the delivery is what fails.

Dear reader,

Every Monday morning a podcast episode appears in my podcast app. It runs about 35 minutes, nobody else can listen to it, and it was made overnight out of my own weekly intelligence brief.

The brief is a curated summary of the week that is relevant to the questions I am actually asking. It reads 71 sources for me, podcasts, newsletters, research and funder announcements, and it scores all of it against my own positioning, my goals, and the handful of things I am working through at the moment. That last part is what makes it worth having. Plenty of tools will summarise the news. This one knows whose desk it lands on.

It used to be a Google Doc, and I was not reading it. For months. Some weeks I opened the email and read the first paragraph, and most weeks I did not open it at all.

Now it arrives in my ears instead, on its own, and I listen to it on a run, driving to pick up the kids, biking to my co-working space, or while I am making dinner.

The whole thing runs on an agentic AI operating system I have been building for months: the context files that hold my positioning and my live work, the routines that run on a schedule, and the connections into the tools underneath. That system is why the rewrite step in here works at all, since the AI doing the rewriting already knows what I care about and who I am writing for.

What took the real work was the rewrite, and that part transfers to anything long you are not getting to. So I wrote down what I did at each step, including the places I got it wrong.

I hope that you will find this useful.



With gratitude,

Erica

Principal, Layer Advisory Services

layeradvisory.com

What this is

A five step path from "I have a long document I keep meaning to read" to "it arrives in my podcast app on its own." It is written for the case where the content is already good and the delivery is what fails.

The centre of it is Step 2, the rewrite spec. That is the set of instructions that turns something written for eyes into something written for ears. You can use that step on its own and never build the rest.

There is a companion markdown file alongside this PDF, `personalized-podcast-companion.md`. You can hand that straight to your AI and it will work through every step with you and do the building, so you do not have to drive any of it yourself.

Who it's for

You already have something long that you are not reading: a weekly brief, a research digest, a board pack, meeting notes, a newsletter you subscribe to and archive unread. You have tried making it shorter and it lost the thing that made it worth having.

You do not need to be technical, and I am not a developer. Nothing here requires you to write code. The last step assumes you are working in an agentic AI system that can run something on a schedule and reach the tools underneath, which today means Claude Code, Claude Cowork, Codex, or something similar.

Before you start

Three things to have in front of you.

- **One real example of the document**, the whole thing rather than a summary of it. You cannot write a rewrite spec against a description.
- **A rough word count**. Divide by 155 for the minutes. My 5,545 word script runs 35 minutes.
- **Somewhere in your week it will go**. Mine is during my daily runs, while I am making meals, or picking the kids up from school.

How to read it

Read it through once, then do Step 2 properly. Steps 4 and 5 are plumbing and can wait a week.

Each step has the same shape: what you end up with, why it matters, what I actually did, how you would do it, the AI skill underneath it, and what to watch for.

Get the script right

STEP 1

Work out whether length is really the problem

OUTCOME

A clear decision about whether to cut the document or change how it reaches you.

WHY IT MATTERS

These two problems look identical from the outside and have opposite fixes. If the document is genuinely bloated, shortening it is right, and a podcast only makes the bloat audible. If the length is carrying the value, shortening it destroys the thing you built.

IN PRACTICE

I nearly cut my brief three separate times. What stopped me was asking what I would lose specifically, rather than in principle. The answer was the connective reasoning: the part where a source gets held long enough to say what it means for my clients, and the part where three sources get compared and a pattern gets named. A summary cannot do either, and between them they were most of the word count. The week I built this, the most valuable thing in the brief was that three unconnected people had arrived at the same conclusion about structuring context for AI, and you cannot see that in a bullet list.

HOW YOU'D DO IT

Open the document and mark every passage that would not survive a 50% cut. Then ask, of each one, whether you would miss it. If most of what would go is restatement and hedging, cut the document and stop here. If what would go is the reasoning that connects things, keep the length and change the delivery.

THE AI SKILL

Asking AI to argue the opposite case. I had Claude make the strongest case for cutting my brief in half and show me the shorter version, so I was reacting to a real artifact rather than defending a preference.

WATCH FOR

Deciding this from memory. If you have not read the document recently you will protect the version you imagine rather than the one that exists.

STEP 2

Rewrite it for the ear

OUTCOME

A plain text script, the same length or longer than the original, that a person could read aloud without stumbling.

WHY IT MATTERS

This is the whole playbook. Every document written to be read carries formatting that is invisible on the page and unbearable in your ears: bullet points, bold labels, section headers, bare URLs, internal shorthand. Put a document straight into any text to speech tool and you will switch it off inside two minutes, then wrongly conclude that the voice was the problem.

IN PRACTICE

My brief has headers like "Evidence added," phrases like "Overlay 3," and a full URL after every source. None of that survives being spoken. What I wrote is a spec that Claude follows every week, and the rule at the top of it is that nothing gets cut. The spoken version comes out longer than the written one, 5,545 words against 3,750, because speech needs connective tissue that a page does not.

One thing I found by accident: my brief already produced a one paragraph summary naming the most important thing that week, which had been sitting there unused for months. It makes a good cold open. Look for something similar in your own document before writing anything new.

HOW YOU'D DO IT

Give AI your document and these instructions, adjusted for what yours contains.

- Nothing is cut. Every item, every piece of evidence, every caveat appears.
- No markdown. No headers, bullets, bold, or numbered lists. Prose only.
- No URLs read aloud. Say "it is linked in the original" if a source must be pointed at.
- No section labels. "Overlay three" means nothing spoken. Say what the section is about.
- Numbers spoken the way a person says them. "Seventy six percent," not "76%."
- Attribution in line and up front: "Nate Jones, writing this week, argued that..." A trailing parenthetical does not survive being spoken.
- Shorter sentences than the written version, one idea each, since a listener cannot re-read.
- Signpost every transition, because there are no visual breaks. "That is the pattern in the tools. The regulation side looks different."
- Repeat a name or number once when it carries weight. Redundancy that reads as clumsy on the page is what makes audio followable.
- Open on the most important thing, in the first sentence.
- Second person if it is for you alone.
- Output plain text with a blank line between paragraphs, no title line and no notes.

THE AI SKILL

Writing a spec once and running it many times, rather than prompting from scratch each week. The spec is the asset. Mine sits in a file that a scheduled routine reads every Monday, so the quality does not depend on how well I happen to phrase the request that day.

WATCH FOR

Accepting the first script without reading it. Read the whole thing once, and every place you stumble is a place the spec needs another line. Mine took three passes before I stopped finding things.

Get it into your ears

STEP 3

Choose a voice and know what it costs

OUTCOME

An audio file, and a cost per episode you have worked out rather than assumed.

WHY IT MATTERS

Voice quality matters less than you would expect and cost matters more, since this runs every week forever. A 35 minute episode is a lot of characters, and pricing in this category is quoted in units that are easy to misread.

IN PRACTICE

I use Google's Gemini text to speech, which comes to about 96 cents an episode, so roughly \$4.20 a month. You connect it with an API key from Google AI Studio, which takes a couple of minutes. I picked a professional voice rather than cloning my own, since listening to myself read me my own brief was strange.

I had planned to use ElevenLabs and my first estimate was about \$5 a month, which was wrong by a factor of four. Their entry plan is 30,000 *credits* a month rather than characters, and the model I would have used spends half a credit per character, so my 33,396 character script needs roughly 72,000 credits a month. The cheap plan would have meant cutting more than half the brief, which is the one thing I had already refused to do.

HOW YOU'D DO IT

Count the characters in your script, not the words, and multiply by the provider's per character rate and then by 4.33 for a monthly figure. Then start free: every Mac has a built in voice, and rendering one episode with it costs nothing and proves the pipeline works before you spend anything.

THE AI SKILL

Knowing which questions AI is bad at. Arithmetic is the one place where you cannot tell by reading whether the answer is right, so it is the thing to recompute rather than skim. I asked for a cost estimate, got a plausible number, repeated it, and was wrong.

WATCH FOR

Paying before you have heard a full episode. A 30 second sample tells you nothing about whether you can tolerate a voice for 35 minutes.

STEP 4

Put it somewhere your podcast app can reach

OUTCOME

A private feed you subscribe to once, that new episodes appear in on their own.

WHY IT MATTERS

Any step that needs you to move a file by hand will stop happening. The point of a podcast app is that it checks for you.

IN PRACTICE

Mine is about as simple as this gets, and that is the appealing part. A podcast is a folder of audio files on the internet plus one text file listing them. There is no podcast host, no account, and no database anywhere in it.

Concretely: the finished audio is copied into a folder on a small private website, and a script rewrites an RSS file in the same folder to add the new episode, with its title, its length, and where the audio sits. The whole folder is published in one command. My podcast app checks that RSS file on its own schedule, sees an episode it has not got, and downloads it. I subscribed once, months ago, and have not touched it since.

Two details worth copying. Everything sits behind a long random web address, so the address itself acts as the password. And the front page of the site returns nothing at all, so there is no page anywhere listing what is there.

HOW YOU'D DO IT

Ask your AI to set up a small static site and generate the RSS file for you. It is a short job and the format has not changed in twenty years. Then add the feed address to your podcast app by URL. Most Android apps take a URL directly. Apple Podcasts on iPhone has no add by URL option at all, so add it on a Mac and let it sync, or use an app that supports it.

THE AI SKILL

Describing the outcome and letting AI choose the parts. I did not specify a host or an RSS library. I said I wanted a private feed my podcast app could subscribe to, and reviewed what came back.

WATCH FOR

Two things I hit. If your document contains client or commercial detail, use an unguessable address so other people cannot find it. And do not block crawlers with a blanket robots file, because podcast apps fetch your feed as bots and one over cautious line there will stop your feed being readable by anything at all.

STEP 5

Make it run without you

OUTCOME

A scheduled task that produces and publishes the episode.

WHY IT MATTERS

A pipeline you run by hand is a chore you will do twice. It only becomes worth having once it arrives without being asked for.

IN PRACTICE

Mine runs at 9:15 on Monday, just after the routine that emails me the brief. It reads that week's document, writes the script, renders the audio, publishes it, and writes one line to a log. If the brief did not run that week it does nothing, rather than sending me a second alert about a failure I already know about.

It runs now without me approving anything, which took a while to earn. I read the first few scripts closely and kept adding rules to the spec every time I found a sentence that would have sounded wrong out loud, and once those instructions were right the whole thing stopped needing me. A full episode takes about 21 minutes to render, which caught me out the first time, so I wrote that into my own notes to save future me diagnosing a slow run as a broken one.

HOW YOU'D DO IT

Write the steps down as a runbook first, in the order they happen, including what should happen when the input is missing. Then point a scheduled task at that runbook. Put a cost limit in the runbook itself, in writing, so it is enforced on every run rather than remembered once. Mine says one AI call, no sub agents, no web searches.

THE AI SKILL

Writing instructions for an agent that runs unattended. The difference from a normal prompt is that you have to say what happens when things are missing or already done, since nobody is watching to make a judgment call.

WATCH FOR

Trusting a success message. Check the actual artifact: fetch the published file, confirm the length, confirm the feed lists it. I lost twenty minutes to a web address that had been typed with one character missing. The error message was real, my explanation for it was invented, and I confidently started fixing the wrong thing.

What I'd do first

If you only do one thing from this, do Step 2. Take the document you are not reading, hand it to AI with the rewrite spec, and read what comes back. You will know within a page whether this is worth building, and the script is useful on its own even if you never narrate it

The order I would go in:

- Open the document and mark what a 50% cut would destroy. If it is mostly restatement, cut it instead and stop there.
- Hand the document and the Step 2 spec to your AI, then read the script that comes back. Every place you stumble is a line the spec still needs.
- Render it with the free voice already on your computer, and listen to the whole thing before paying for anything.
- Only once you have listened to one all the way through, set up the hosting and the schedule.

That last condition is the real test, and it is why I have written the steps in this order. The pipeline is the easy part, and it is worth nothing if the script is wrong.

If you would rather not do any of this by hand, give your AI the companion file that came with this playbook, `personalized-podcast-companion.md`, and ask it to work through the steps with you.

/erica-layer

AI ON PURPOSE

Build the operating system underneath.

The podcast was the easy part. What made it possible is that the brief behind it was already written against my positioning, my goals, and the questions I am working through, because I had built an AI operating system that knows them.

TEMPO is my six-week cohort for non-technical people who want to build exactly that: the context files, the daily routines, the knowledge base, the loops that hold AI to your standards, and the automations that run without you. You build it on your own real work, and you leave owning it.

Everything in this playbook came out of that system.

layeradvisory.com/tempo

I'm Erica Layer, an AI-native fractional COO. I help purpose-driven organizations build the human and AI systems their work runs on, and I teach people to build their own.

layeradvisory.com · aionpurpose.substack.com