HOW WE WORK

The work behind the result.

People spend a large part of their week on work a machine can now do well. Every hour that goes into chasing, copying, retyping and answering the same question again is an hour that does not go into winning work. Our job is to take that work off your team and give the time back to the things that grow the business.

How we do that depends on which of the four problems you have, so the process is different for each one. All four are set out in full below.

Three things hold on every job.

A number you rely on never comes from an AI

Anything you would put in front of a customer, a board or an accountant is calculated in ordinary code that gives the same answer every time. A model can draft the words around a figure. It never produces the figure.

A person approves anything that leaves the building

Sending a message, spending money, publishing, committing you to something. Those stay with a human by default. The system prepares the work and then waits.

You own what we build

The code, the accounts and the data are yours, in your name, from the start. If you stop working with us you keep all of it and it keeps running.

How each service line runs.

Same standards, four different jobs. Each one sets out the sequence, what we need from you, and what you have in your hands when a stage closes. We put no times against them, because the honest answer depends on your business, and a number invented to look confident is worth nothing to you.

AI consulting and training

AI in untrained hands is a cost with nothing to show for it. The licences get bought, a few people try it, the rest go back to how they worked before, and the bill arrives anyway. The limit is almost never the tool. It is how far along your people are, and they are rarely all in the same place.

1. Measure where you actually are
2. Watch the real work
3. Rank what is worth doing

GATE

4. Train each level on your own work
5. Set the rules
6. Come back and check it stuck

The sequence, start to finish. We stop once and wait for your sign-off.

WE STOP HERE

You sign the plan

Nothing is designed or delivered until you have the ranked plan in writing and have agreed it. If the honest answer is that AI would not pay back here, this is where we tell you.

01 Measure where you actually are

We assess two groups separately, because they are usually far apart: what your leadership understands about what AI can and cannot do, and what the people doing the daily work can actually get out of it.

YOU BRING
- Access to both groups
- Honesty about what has already been tried

YOU GET
- A plain reading of where each group sits
- The gap between the two, named

02 Watch the real work

We sit with people while they do their job and look for the places where AI would genuinely help. Some of what looks automatable is not, and we say so.

YOU BRING
- Time with the people who do the work, not only those who own it

YOU GET
- A map of where the hours actually go

03 Rank what is worth doing

A short list, in order, with the effort and the running cost against each one. Things that are not worth doing come off the list here, rather than after you have paid for them.

YOU GET
- A ranked plan in plain English
- The expected running cost of each item

04 Train each level on your own work

GATE BEFORE THIS

Separate sessions for separate levels, built around your real documents and your real tasks. Generic training does not survive contact with a Monday morning.

YOU BRING
- Real examples of your work we may use
- People in the room

YOU GET
- Training built from your material
- Something each person can use the same day

05 Set the rules

What may be put into a model and what may not, who signs off what, and a spend limit that does not creep. Written down, so it outlives the people who were in the room.

YOU GET

- A short usage policy your team will actually read
- A predictable monthly ceiling

06 Come back and check it stuck

We return and find out what people are still using and what they quietly abandoned. Abandonment is the normal outcome of training, and fixing it is the point of this stage.

YOU GET

- An honest read on adoption
- The abandoned things either fixed or dropped

Process automation

A messy process is usually not a technology problem. It is that nobody has written down who does what, what counts as correct, and what is meant to happen when it breaks. Automating a process nobody has agreed on only makes the wrong thing happen faster.

1. Map it as it really runs
2. Find the owner and the failure paths
3. Agree what correct looks like
4. Decide what stays human

GATE

5. Build it against the examples
6. Test the awkward cases

GATE

7. Fit it into what you already run
8. Hand it to a named person

The sequence, start to finish. We stop twice and wait for your sign-off.

WE STOP HERE

You sign the spec

No build starts until the spec and the examples are agreed in writing. This is the point at which scope is cheap to change, and the last point at which it is.

WE STOP HERE

You accept it

You test it on real work, against the examples you signed. Nothing goes live on anything you have not accepted.

01 Map it as it really runs

With the person who does the work, not the person who owns it. Including the steps that live only in somebody's head, and the exception everyone has stopped noticing.

YOU BRING
- An hour with whoever actually does it
- Permission to see the real cases

YOU GET
- A written map of the process as it is today

02 Find the owner and the failure paths

Who is accountable when this goes wrong, what happens today when it does, and who notices. If nobody notices, that is the first thing we fix.

YOU GET
- A named owner
- The failure paths written down

03 Agree what correct looks like

We collect real examples, the awkward ones included, and agree the right answer for each. Those become the test the finished system has to pass, so correct stops being a matter of opinion.

YOU BRING
- Real examples, the messy ones included
- A decision on the edge cases

YOU GET
- A signed spec
- The set of examples we will be judged against

04 Decide what stays human

Anything that sends money, sends a message or commits the business goes to a person for approval unless you tell us otherwise. We agree that line before anything is built, not after an incident.

YOU GET
- An agreed line between automatic and approved

05 Build it against the examples

GATE BEFORE THIS

The system is built to pass the examples you signed off. Progress is measured as how many it passes, so there is no argument about whether it is finished.

YOU BRING
- Access to the systems it needs
- Somebody to ask when a case is unclear

06 Test the awkward cases

Empty input, a system it depends on being down, the same job running twice, permission refused, a rate limit. Anything that sends or spends has to be safe to repeat. Most of the real work is here.

YOU GET
- The result of every case, pass or fail, in writing

07 Fit it into what you already run

GATE BEFORE THIS

It has to live inside the tools your team already opens. A system people have to remember to visit is a system that gets forgotten.

YOU BRING
- The systems it must sit alongside

08 Hand it to a named person

Someone who owns it, alerts we have proved reach a human, and a written runbook for when something needs doing. We do not close a job with nobody holding it.

YOU BRING
- Somebody to own it

YOU GET
- A runbook
- Alerts tested in front of you
- The code and the accounts

AI agent solutions

An agent that invents an answer is worse than no agent, because people believe it. Everything below is arranged around one thing: making it answer from your material, and making it say it does not know when it does not. Getting an agent to refuse well is harder than getting it to reply.

1. Decide what it answers and what it refuses
2. Find where the knowledge really lives
3. Prepare the material so it can be searched

GATE

4. Build the confidence gate first
5. Build the agent behind the gate
6. Test it on real questions

GATE

7. Put it where your people already are
8. Put a person on the flags

The sequence, start to finish. We stop twice and wait for your sign-off.

WE STOP HERE

You sign the scope

The answer list, the refusal list and the material are agreed in writing before anything is built. That document is what the finished agent gets judged against.

WE STOP HERE

You accept it

You see the graded results and try it yourself before a single customer does. It goes live when you say it does.

01 Decide what it answers and what it refuses

The refusal list matters more than the answer list. Prices, promises, medical or legal questions, anything only a person should say. We agree this before anything else.

YOU BRING — A view on what it must never handle

YOU GET — A written scope, including everything it will decline

02 Find where the knowledge really lives

Usually not where people expect. Recorded calls, a shared drive, help desk history, somebody's notes, videos. The best material has often never been written down.

YOU BRING — Access to your material

Somebody who knows where the good answers are

YOU GET — An inventory of what we found and what is missing

03 Prepare the material so it can be searched

Transcribing, cleaning and indexing. This is the largest part of the work and the part nobody sees. The quality of every answer it will ever give is decided here.

YOU GET — Your knowledge in one searchable place, which is useful on its own

04 Build the confidence gate first

GATE BEFORE THIS

Ordinary code, not a model, decides whether your material actually holds a good enough answer. Below that line the model is never asked to make a claim. That is the mechanism behind saying I do not know, and it exists before the agent does.

YOU GET — A refusal you can rely on rather than hope for

05 Build the agent behind the gate

Now the part people picture. It answers in your voice, from your material, and it never gets to answer a question the gate has already stopped.

YOU BRING — Examples of how you would answer, in your own words

06 Test it on real questions

Including the ones it must refuse, the ones written badly, and the ones designed to talk it into something. We grade the answers rather than glance at them.

YOU BRING — Real questions your people actually ask

YOU GET — A graded set of results before it meets a customer

07 Put it where your people already are

GATE BEFORE THIS

Their chat tool, their inbox, your site. Wherever the question already gets asked, rather than somewhere new they have to be persuaded to go.

YOU BRING — Access to the channel it will live in

08 Put a person on the flags

Everything it declined goes to a human. Those answers then go back into the material, so the set of things it cannot handle shrinks every week instead of standing still.

YOU BRING — Somebody to review the flags

YOU GET — A queue of what it could not answer

YOU GET — A system that improves rather than drifts

Digital redesign

A dated site costs you work you never hear about, because the people who bounced off it do not write in to tell you. A site also has two audiences now. People read it, and so do the assistants people increasingly ask before they ever visit. Most sites are built for only the first.

1. Agree the one job the site has to do
2. Look at what happens on the current one
3. Set one direction and sign it off
4. Write it before designing it

GATE

5. Build it
6. Test it on real devices
7. Make it legible to machines as well as people

GATE

8. Launch and hand over

The sequence, start to finish. We stop twice and wait for your sign-off.

WE STOP HERE

You sign the direction

The direction and the words are agreed before the build starts, so what you see at the end is what you approved at the beginning.

WE STOP HERE

You accept it

You walk the finished site on your own phone before it replaces the old one. It launches on your word.

01 Agree the one job the site has to do

Get an enquiry, get a booking, get a phone call. Sites that try to do five things do none of them. Deciding this first is what makes every later decision easy.

YOU BRING — A decision on what a win looks like

YOU GET — One measurable job, written down

02 Look at what happens on the current one

Where people arrive, what they read, where they give up. If nothing is measuring that today, we say so plainly rather than inventing a before figure.

YOU BRING — Access to whatever analytics exist, if any

YOU GET — An honest read on the current site, or a clear note that there is no data

03 Set one direction and sign it off

One direction, agreed before anything is built. Redesigning during a build is the most expensive way to work and the most common.

YOU GET — A direction you have seen and approved

04 Write it before designing it

The words decide the layout, not the other way round. A design built around placeholder text gets rebuilt when the real words arrive, and the real words are always longer.

YOU BRING — Time on the copy, or the decision to have us write it

YOU GET — Approved words for every page

05 Build it

GATE BEFORE THIS

Built to load fast on a phone on a poor connection, because that is where most of your visitors are, and because a slow site loses people before it has said anything.

06 Test it on real devices

Actual phones and actual tablets, not a desktop browser made narrow. Phone first, since that is where the majority of the traffic and the majority of the problems are.

YOU GET

- Results across the devices your visitors really use

07 Make it legible to machines as well as people

Structured so search engines and AI assistants can read what you do, who you serve and how to reach you. More buying decisions now start with a question to an assistant than with a search.

YOU GET

- A site the tools people now ask can quote correctly

08 Launch and hand over

GATE BEFORE THIS

Everything in your name. The domain, the hosting, the code. You should never need our permission to change your own website.

YOU GET

- Ownership of the code and every account
- A short guide to changing it yourself

What you end up holding.

Every stage produces something you keep. None of it is ours to take away, and none of it needs us present to be readable.

- A written map of how the work runs today
- A spec you signed before anything was built
- The real examples we agreed to be judged against
- Test results for every case, pass or fail
- A runbook for whoever ends up owning it
- The code and every account, in your name

When we are the wrong firm.

It is cheaper for both of us to find this out now.

- You want a demo to show a board rather than something that runs on Monday.
- Nobody in the business can be freed up to answer questions about how the work is done.
- The process changes so often that there is nothing stable to build against yet.
- You want the system to send, spend or publish with no human anywhere in the loop.
- You need it live next week. We would rather say no than ship something we have not tested.

What happens if you get in touch.

01

A thirty minute call

You describe what is eating the time. We tell you straight whether it is worth automating, and if it is not, we say so. There is no deck.

02

A scoping note

A short written note naming what we would build, what it would need from you, and what it would cost to run. You keep it whether or not you go ahead.

03

You decide

No pressure, and nothing to sign before you know what you are getting. If the note does not convince you, that is a useful answer too.

Book a thirty minute discovery call.

Pick a time that suits you. We will ask what the work is, who does it now, and what it is costing you in hours. By the end of the call you will know whether there is something here worth doing.

Book a discovery call

https://cal.com/pulsarai/intro-call