Grey Matters
ExploreGreyMatters.ai

CASE STUDY • LEARNING LIBRARY

Building Grey Matters — and Becoming Its Accidental Release Operator

A real website, a steep learning curve and the discovery that human control should not mean doing every technical job.

THE STORY OF BUILDING AND LEARNING

Human-led. AI-enabled.

Grey Matters • October 2026 • Publication edition v1.0

It seemed like the obvious next step

By the time we decided to give Grey Matters a public home, there was already quite a lot to put in it. What had begun as conversations with AI had grown into a book, research work, visualisations and practical projects involving everything from travel and gardens to difficult decisions. The website wasn't meant to be the achievement in its own right. It was meant to make some of those experiences useful to other people.

The idea sounded straightforward enough: build an accessible place where somebody could arrive with an interest, find a real example, try something themselves and, if they wanted to, go deeper. I didn't have a software-development background. That wasn't the point of the exercise. The question was whether someone who knew what they wanted to communicate could work with AI to get it built without first retraining as a developer.

I had already discovered how far an ordinary conversation could take me. Perhaps the website would be another example. It was. But it taught me rather more about what happens *after* an AI has produced a plausible answer.

The easy bit was imagining the website

AI helped turn the idea into a structure: ways of working, projects, case studies, simple prompts for visitors and a research area that allowed people to inspect what we had actually done. It could write copy, propose designs, make changes and spot inconsistencies. The remarkable thing was the range of skills it could bring to a single project.

But a website isn't only a collection of pages. It has to live somewhere. Someone must control its address, its source files, its permissions and the difference between a private draft and what the public actually sees. There are accounts to connect, services to configure and versions to keep straight. And an AI that can confidently describe those activities may not have the access or practical ability to carry them out.

I learned that distinction by becoming the link between systems. When a login failed or a browser session collapsed, I returned to a dashboard. When a workflow needed a technical identifier, I copied it from one place to another. When a tool reported success or failure, I sometimes had to inspect the underlying service to discover which was true.

At this point I was still making the important decisions, which was right. Unfortunately, I was also doing a growing number of jobs that were neither interesting nor a particularly good use of human judgement. I had become the accidental release operator.

> There were moments when I was hanging on by my fingertips. The AI could move from writing a page to code, security settings and deployment procedures in the time it took me to understand the last recommendation. Being able to ask good questions was an advantage; it didn’t magically give me the background to judge every technical answer. Sometimes it felt as though the choices were to trust the AI blindly or admit I was not qualified to lead the project.

Neither choice was satisfactory. I could not become a developer overnight, but I could insist on understanding the purpose, consequences and limits of a decision before approving it. Staying in

the loop meant asking awkward questions, going back over unfamiliar explanations and occasionally saying “no” or “not yet”. That persistence was part of the work, not evidence that the collaboration had somehow failed.

A little less ambition would have helped

Our first substantial mistake was almost flattering: AI made extra capability look so achievable that it was tempting to keep adding it. Visitors needed a way to get in touch. Before long we were exploring structured submissions, an operator workflow, a database, authentication and separate technical environments.

There were reasons for each addition. The problem was the cumulative effect. Every new piece brought permissions, tests and failure possibilities; each attempt to make it safer seemed to require another piece of engineering. Meanwhile, the actual public website was still waiting.

A real publication deadline concentrated the mind. What did a visitor need on day one? A clear website and a working contact route. An email address would do. We kept the more ambitious work for a future need rather than pretending it had to be finished before anybody could see the site.

Grey Matters launched on 22 October 2026. That was a useful reminder that choosing *not* to build something can be a proper project decision. There was no prize for delivering a sophisticated contact system to an audience that hadn't asked for one.

> There was a human cost to reaching that decision. In the early days the freedom to build something new was exhilarating; later, repeated logins, permissions and recoveries could take over evenings and stray into time that was supposed to be a break. I hadn’t set out to become an unpaid administrator of several unfamiliar services. A simpler first release was therefore not just a technical correction. It gave some of my time and attention back to the reason for making Grey Matters in the first place.

A green tick is not the same as a working result

Some of our most instructive moments were technical failures that challenged an apparently sensible conclusion. One build produced a different file fingerprint even though the visible content had not changed: the packaging process had introduced variable information. A separate release really did go live, but a check expected a different response from the hosting service and announced failure. Neither incident was solved by trusting the first message on the screen.

The lesson wasn't that automated checks were useless. They were vital. It was that a check is only as good as what it actually checks. We had to look at the real files, the real service and the intended result. A confident AI summary, an alarming error or a reassuring green tick were all things to investigate, not verdicts to accept blindly.

The human contribution here was not to understand every line of code. It was to keep asking: Is this the result we intended? What evidence shows that? What would we do if it isn't?

> When I was out of my depth, the most useful response was not to become fluent in every technical detail overnight. It was to ask the AI to slow down, explain the decision in ordinary language, distinguish what it knew from what it assumed and show what would happen if the recommendation were wrong. I sometimes needed the same point explained more than once. That wasn't a reason to hand over control; it was how I learned enough to exercise it. And where I couldn't judge a risk, the right answer was to pause or seek proper expertise, not bluff.

Moving the human back to the right job

After launch, we began trying a different division of labour. An AI agent, working through approved and limited procedures, could prepare and check a particular website change, test it away from the public site and carry out the repeatable steps towards release. I retained the decision about what we wanted, what was acceptable, what access was granted and when something could go public.

There was a moment that captured why this mattered. A release encountered a permissions rule and stopped. The agent didn't try to work around it. It reported the boundary and waited for the authorised human decision. That was better collaboration than a clever workaround would have been.

We also learned to record decisions outside our conversations, work against a known approved version, make changes in manageable pieces and have the actual result challenged. The point was not to turn every ordinary question into an elaborate process. It was to apply more care when the consequences became real.

The Learning Library: another real test

On 30 October, we put the same approach to work publishing the Learning Library's first two introductions and two downloadable resources. This was no longer a hypothetical release process. It was something intended for visitors to read and use.

The human part was unmistakable. On the actual pages, we noticed an overly aggressive crop of the panoramic imagery, a pale visual rhythm and text broken into a succession of isolated one-sentence paragraphs. Those weren't problems a source-code check alone could settle. They were questions of meaning, readability and taste, judged by looking at the site on a laptop and a phone. The approved words were preserved while their presentation was improved.

The engineering controls also earned their place. One staging run stopped before uploading because a required PDF-checking utility wasn't present. A subsequent run found that the hosted runner's Python version interpreted a robots-file check differently from the local environment. The problems were fixed and tested before the public release. The failures cost time, but the stop conditions did what they were supposed to do: they prevented an uncertain package from moving into production.

The Learning Library release was then checked in protected staging, approved and promoted to the live site. A real, human-reviewed publication was the result. Yet it would be misleading to claim that the whole process had become quick or effortless. The routine technical operations had moved

away from the human, but there were still many handoffs, checks and approvals. We had changed the *kind* of effort more clearly than we had reduced its total amount.

The next lesson is about proportion

This is where we have reached, rather than where the story ends. For a substantial or irreversible change, careful checks, recovery arrangements and explicit approval are essential. But two ordinary publication PDFs shouldn't need a fresh engineering expedition every time.

We are now considering a smaller, repeatable route for routine document updates: agree the editorial work, identify the exact files, inspect a private version, approve the public release and verify what visitors can actually see. The more detailed machinery can sit underneath that sequence. It still has to be tested; it isn't an achievement we can claim yet.

> That includes protecting the human’s attention. The very freedom that made it possible to work on ideas with a book and a phone while away on holiday also made it surprisingly easy for a technical problem to follow me there. A few minutes exploring an image could turn into another session recovering a failed connection or checking a deployment. The work was absorbing, but absorbing is not always the same as worthwhile. Part of learning to operate the site was learning when to put the tools down.

I also want to be clear about the limits of the story. This is one person's experience with one website and its particular tools. We haven't produced a reliable measurement of time saved or shown that the same method will work equally well for everybody else. Nor should the successful publication of our research be mistaken for proof of the scientific ideas presented there.

What has changed is the question I ask of human–AI collaboration. It isn't merely, “Can the AI do this?” It is, “Can we make something worthwhile, know what actually happened and keep the human involved where the human adds value?”

That feels like a much more useful question. And it's one I wish I'd asked a little earlier, before I learned quite so much about deployment dashboards.

If you're thinking of building something yourself

Start with the thing you want other people to use, not the collection of services you could install. Keep the first version small. Look at it yourself. Ask the AI to find problems rather than congratulate itself. Record what you have agreed, and know how to get back to the last working version. If the AI is moving faster than you can understand, ask it to explain the actual decision and its consequences. You do not need to know everything; you do need to know enough to stay responsibly involved — and when to ask for expert help. Most of all, don’t confuse human leadership with having to perform every technical operation yourself.

The companion guide, *Build & Operate with AI: A Practical Starter Playbook*, translates these experiences into steps you can adapt to a small website or another public-facing project. You can read either publication first.

Grey Matters — Human-led. AI-enabled.