

AI-Assisted Modernisation Pilot

One subsystem off your modernisation roadmap, delivered, with before-and-after numbers you can put in front of a board. Four to six weeks. This is where you find out what AI-assisted delivery actually does on your code, rather than in someone else's demo.

WHO IT IS FOR

Teams that already have a roadmap, mine or someone else's, and want evidence that AI-assisted delivery works on their own brownfield code before committing a programme budget to it. Whatever the language: most of my pilots are .NET, but the method is the same on anything with a compiler and a nervous owner. Six weeks on one subsystem answers the question either way.

WHAT GETS DELIVERED

- The first phase of the roadmap, done. Usually a framework uplift, a service pulled out behind an API, or a performance rescue on the path everyone complains about. We pick it together in week one.
- A characterisation test harness, built before I change anything. It pins down what the code currently does, so every change gets checked against real behaviour rather than good intentions. The harness stays with you.
- The before-and-after measurements, agreed in week one so nobody is marking their own homework at the end. Typical set below.
- A write-up of the working method: tooling setup, guardrails, review gates, the prompts that earned their keep. Your developers shadow the work as it happens, which honestly is where most of the transfer takes place.

METRICS AGREED UP FRONT

Dimension	Typical measures
Delivery	Cycle time for a change in the subsystem; build and deploy time; how long a new developer needs before making a safe change
Quality	Test coverage on the subsystem, usually from near zero; defects during and after; static analysis findings closed
Performance	Response times and batch durations on the paths the business actually complains about
Risk	End-of-life components retired; areas that were one-person-only now documented and tested

WHAT IT NEEDS FROM YOU

- Write access to a branch and a route into test environments
- A named developer or pair to shadow the work; part-time is fine
- Sign-off on the week-one scope and the metric set

GUARDRAILS AS STANDARD

- Characterisation tests before anything changes
- Locally hosted models where code cannot leave the building
- A human review on every merge, no exceptions

Why a pilot rather than a programme. Six weeks and one subsystem is enough to prove the method or kill it. If the numbers do not stack up, you stop, and you keep the harness, the finished subsystem and the write-up. Either way you end up with real figures for the business case instead of vendor slides. Fixed scope, deliverable-based, outside IR35.

Who is doing the work. Darren Bowles, Epona Solutions. Twenty-five years of legacy migrations, from RPG/400 and mainframe logic extraction through to AI-assisted brownfield .NET. I use the agentic tools daily, and I have spent a career being the person called when legacy changes go wrong, so I know where they bite.