



SERVICE BRIEF

Brownfield and Greenfield Code Development

PLC and SCADA software written to your standard, or ours, and proven before it reaches site.



THE CHALLENGE

Software your engineers can own

Code nobody fully understands

Years of changes by different engineers leave logic that works but cannot be explained, so every modification feels like a gamble with production.

Your standard, not theirs

You have a coding standard and a block library for good reason. Suppliers who ignore them hand over software your team cannot maintain.

Faults found on site

Logic errors found at commissioning cost the most: idle crews, delayed production and pressure to fix things live.

Documents that drift

Specifications, I/O lists and alarm lists fall out of step with the code, and no longer describe what actually runs.

WHAT WE OFFER

Six ways we develop your software

OPTION 01

Your standard, your blocks

We write to your company coding standard, naming rules and block library, so the result looks like your own engineers wrote it. Your library blocks are reused, not copied and altered.

OPTION 02

Our standard

Where you have no standard, we use ours: a consistent structure of equipment modules, state-based sequences, alarms and interlocks, laid out the same way on every project.

OPTION 03

A standard built with you

We create a standard with you: structure, naming, block library, templates and coding rules, documented so every future supplier and engineer follows it too.

OPTION 04

Generated from your examples

Give us one well-written example of a motor, valve or sequence. We generate the rest of the project in the same form, so code is consistent and quicker to review.

OPTION 05

Brownfield: understand, then change

We analyse your existing code first: structure, sequences, I/O, alarms and interlocks. Missing documents are recovered from it, and changes are made under control.

OPTION 06

Greenfield: specification to code

New systems start from an agreed functional specification. The software is built from it in a clear structure, so every function traces back to a requirement.

BROWNFIELD AND GREENFIELD

Two starting points, one discipline

Brownfield: an existing plant

- A restorable backup of what runs today, taken before anything changes
- Code analysed and I/O, alarm and cause-and-effect lists recovered from it
- Changes written in the style and structure already there
- Each change tested in simulation against today's behaviour
- Cut-over planned around your outages, with a rollback plan

Greenfield: a new system

- Functional design specification agreed before code is written
- Standard chosen up front: yours, ours or a new one
- Consistent code generated from the plant's equipment list
- Software proven against a simulated plant from the first build
- Full document set produced alongside the code, not after it

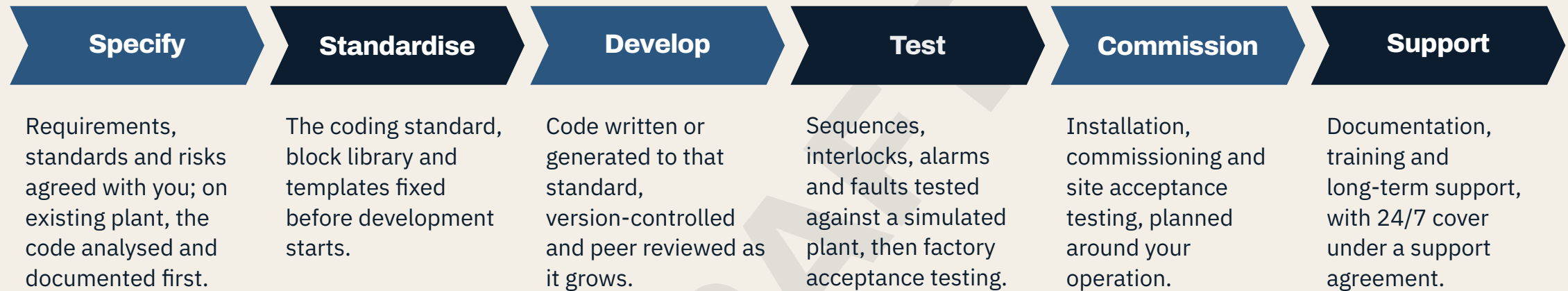
COVERAGE

What we develop

Area	What we develop	How it is proven
PLC software	IEC 61131-3 code in ladder, function block diagram and structured text; statement list where the existing program uses it	Compiled clean, reviewed, sequence and interlock tests
Safety programs	Safety functions on safety PLCs and relays, written to your safety requirements specification	Safety requirements tests, independently reviewed
HMI and SCADA	Screens, alarms, trends, reports and operator commands, consistent with the PLC structure	Tested against the PLC running in simulation
Drives and devices	Drive, instrument, vision and robot interfaces, with parameters archived with the project	Interface tests and parameter records
Communications and data	PROFINET, EtherNet/IP, Modbus and OPC UA links, historians, reports and MES or ERP interfaces	Point-by-point communications tests
Version control	Every change committed with its reason, reviewed and traceable from the first line to handover	Change history handed over with the source

HOW WE DELIVER

From requirement to supported system

**WHERE THE LINE IS**

Simulation proves logic against a model of your plant, within that model. It never replaces tests on the real equipment, and it is never safety acceptance. Process limits and safety targets come from your process design and hazard studies; we implement, test and record them.

WHAT YOU RECEIVE

Software and the documents that explain it

✓ Functional design specification

✓ Coding standard and block library

✓ Alarm list and HMI interface

✓ Source code with its change history

✓ Training and fault method statements

✓ Software design specification

✓ I/O list and cause-and-effect

✓ FAT and SAT protocols and records

✓ As-built drawings and evidence index

HOW WE PROVE IT

- Every function traced from the specification to its code and its test
- Code reviewed by an engineer who did not write it
- Sequence, interlock and alarm tests run against a simulated plant before factory acceptance
- Deviations logged with their fix and retest; nothing closed silently
- On running plant, every change staged from offline to simulation to live, each step signed off

GET IN TOUCH

Talk to us about your software

enquiries@matrixsystems.co.uk

0121 240 7454

www.matrixsystems.co.uk

ISO 9001 certified • Cyber Essentials • CHAS • Constructionline • SafeContractor

