Sample Project Input

Short website version

Website sample	Simplified fictional example based on a real legacy medical device software scenario.

This sample shows the kind of starting information used to scope a remediation engagement.

Scenario

- Portable patient monitor for non-clinical observation in normal daily environment
- 3-lead ECG at 250 Hz, SpO2 and pulse at 1 Hz
- 72-hour battery operation, automatic recording at power-on
- Data stored in external flash and later downloaded to a C++ / Qt PC application
- Physician uses the PC software for visualization and CSV export only

Key technical constraints

Area	Current condition
Device platform	Cortex-M MCU, battery powered, USB-to-serial with galvanic isolation
Sensors	Third-party certified modules over I2C, polled by the MCU
Storage	External flash on main board, ring-buffer concept, multiple sessions
Safety-relevant product behavior	Use during charging is prohibited. Charger connection stops recording immediately.
Timestamping	MCU-generated timestamps, RTC present, current UTC time synchronized on PC connection
PC software	C++ / Qt, file-based import, visualization, CSV export
Verification baseline	Mostly manual tests, limited unit tests, partial legacy documentation

Why this turns into remediation work

- Architecture exists in code but is not explicit enough for efficient maintenance or evidence generation.
- Core lifecycle behavior is tied to power, memory, storage, and communication conditions.
- Testing is not strong enough to support confident refactoring without additional work.
- The client needs a structured view of what exists now, what is missing, and how to improve it without a rewrite.