

AI-Assisted Software Engineering

Orchestrated engineering environments for SMEs

The human defines the objective and accepts the result. An orchestrated AI engineering environment performs the engineering work in between.

THE BUSINESS PROBLEM

Custom software is often too expensive, too slow or too difficult to staff for small companies. AI-assisted engineering can move that economic boundary when it is embedded in a controlled engineering process without removing human ownership of the result.

THE TARGET MODEL

Not a single coding assistant, but a Scrum-oriented engineering process: prioritised work, short iterations, specialised engineering functions, defined quality gates, working increments and human acceptance.

SCELARIS CAPABILITY

Scelar designs and implements orchestrated AI-assisted software engineering environments for SMEs.

HUMAN DIRECTION

Business goals
Priorities
Acceptance criteria



ENGINEERED RESULT

Working software
Verification evidence
Documentation

HUMAN OWNERSHIP

The human remains Product Owner and acceptance authority. AI executes the engineering workflow.

ENGINEERING DISCIPLINE

Success is reproducibly working, tested and documented software - not generated code volume.

OWN ENGINEERING USE

We operate and expand the approach in our own engineering environment.

CURRENT STATUS

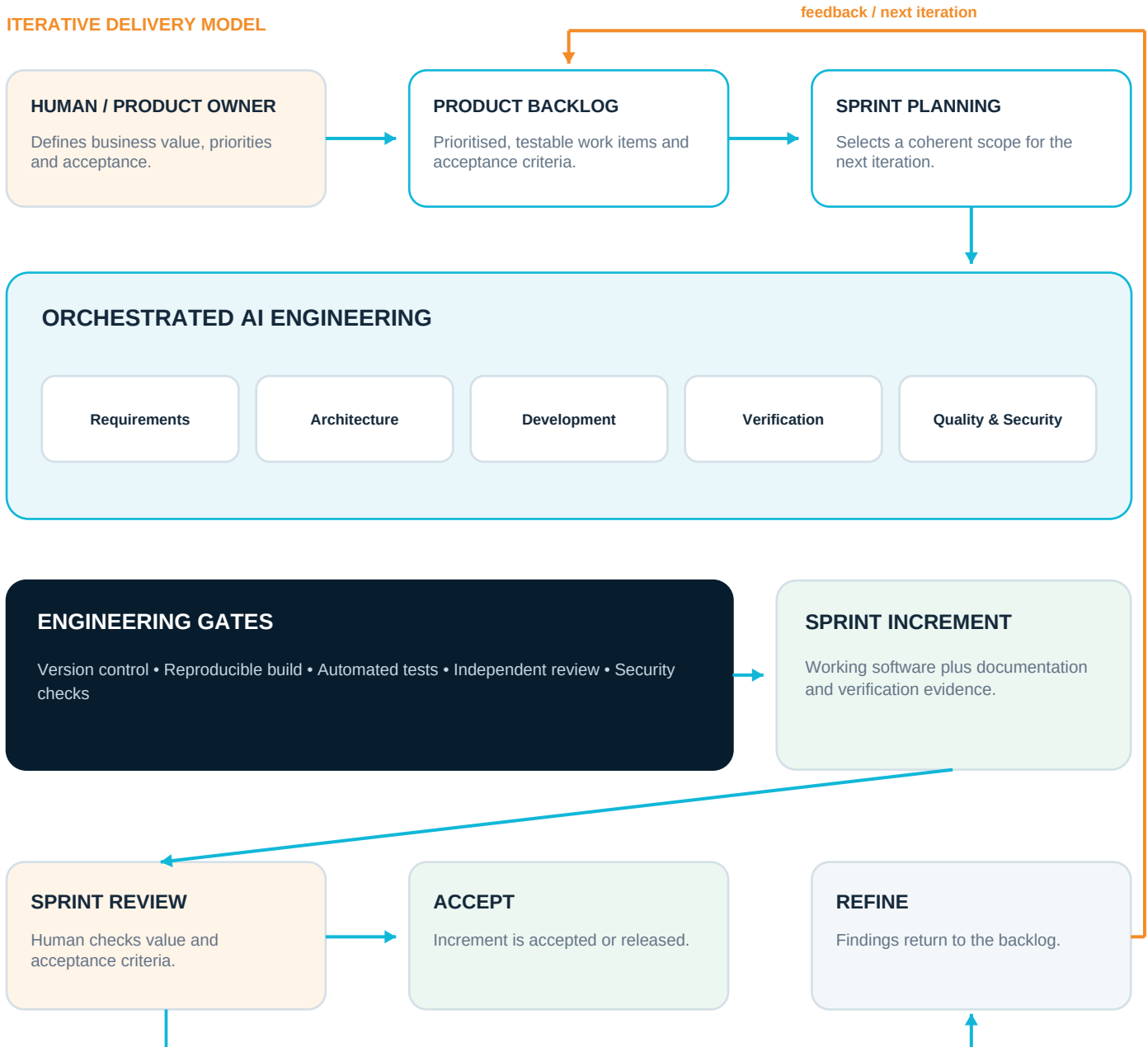
Scelar already uses and continuously expands AI-assisted engineering in its own development environment. The integrated end-to-end workflow described here is the next stage: consolidating these practices into a reproducible engineering process with defined quality gates and human acceptance.

OPERATING MODEL - DELIBERATELY ABSTRACTED

A Scrum-oriented AI engineering cycle

Visible engineering discipline and control points; internal orchestration remains solution know-how.

ITERATIVE DELIVERY MODEL



VISIBLE ENGINEERING DISCIPLINE

Concrete outputs and gate results are visible and auditable. The public model deliberately shows the engineering discipline and control points while model selection, internal topology, prompts, context handling, routing, decision logic and orchestration mechanisms remain Scelar solution know-how.

What matters: working software, reproducible evidence, human acceptance.