

Four early signals in complex engineering programmes

Four perspectives for seeing early how well a programme is creating the conditions for the overall system to come together.

End-to-end Focus

Structure

Evidence

Capabilities

Does the programme organise around what the overall system needs to achieve?

- Is that outcome clear and aligned with what the customer actually needs?
- Are programme priorities set around it?
- Do programme leadership and teams use it to guide decisions across the project?
- Are dependencies addressed before late phases force them into view?

WHEN THIS WORKS

The programme converges on what is needed, and later phases mostly confirm the direction.

WHEN IT DOES NOT

The parts progress, but the overall system falls short of what is needed, forcing difficult late changes and compromises.

Does the organisational structure support how the overall system needs to come together?

- What shapes the programme structure more: how the system needs to come together, or the boundaries between disciplines and subsystems?
- Do decision rights sit close to the relevant knowledge, with clear guardrails for alignment?
- Do responsibilities and decision paths adapt as the programme changes?

For example:

If Systems Engineering owns system coherence, which architecture and interface decisions can it make without escalation?

WHEN THIS WORKS

Decisions stay close to the relevant knowledge, with coordination focused where alignment is needed.

WHEN IT DOES NOT

Decisions escalate, coordination expands, and scarce expert time compensates for weak structure.

Does the programme have reliable evidence of where the system really stands?

- Is status based on demonstrated functionality, not just completed work and milestones?
- Are key system functions demonstrated before the full system is complete?
 - In a representative operational environment?
- Do new system-level results change priorities, commitments and decisions?

WHEN THIS WORKS

Problems surface while there is still time to act, keeping commitments grounded in evidence.

WHEN IT DOES NOT

Gaps surface late, when change is expensive and rework threatens commitments.

Has the programme built the capabilities its success depends on?

- What does this programme repeatedly need to do well?
 - Are those capabilities built across leadership, teams and disciplines?
 - Are they embedded in everyday work?

For example:

If the system has to work in the customer environment, is enabling that a shared programme capability, or does one integration role repeatedly have to fight for the needed contributions?

WHEN THIS WORKS

The programme gets progressively better at the things its success depends on, using time and expertise more effectively.

WHEN IT DOES NOT

The same problems keep returning, consuming scarce time and resources instead of building capability.

What do you see in your programme?

These four perspectives are connected. Weak evidence can make commitments unreliable. Unclear decision rights can turn dependencies into coordination overhead. Missing programme capabilities can leave a few experienced people repeatedly holding things together.

The useful question is where unnecessary difficulty is emerging, and what would make the biggest difference to how the overall system comes together.

- Where does the programme repeatedly struggle despite capable people doing good work?
- Where does progress depend heavily on individual people holding things together?
- What becomes visible late that would be valuable to know earlier?
- Which of the four perspectives would be worth looking at more closely?

Markus Hager

Independent Consultant

Working where engineering programmes become difficult to bring together.

insidecomplexity.today