



HOW YOU STAY IN CONTROL

You keep authority over your product. We carry the delivery.

For teams that have been let down by a vendor before: what you decide, what every change must pass, what you keep, and how an engagement ends cleanly.

THE CONTROL POINTS

01 You decide

Direction, priorities, scope changes and production releases.

02 Named ownership

A named owner on each side for every agreed outcome.

03 Agreed access

Granted by you, limited to the work, revocable by you.

04 Gates for every change

Shown working, tested, released with an agreed recovery path, accepted by you.

05 What you keep

Code, infrastructure, runbooks and decisions, in your systems.

GATES, CHANGES AND ESCALATION

Nothing reaches production, and nothing changes scope, without you

EVERY CHANGE PASSES



SCOPE AND CHANGE APPROVAL

- Every change request is written down
- Its effect on scope, timing and cost is stated
- Work starts only after your approval
- Nothing is assumed beyond what is agreed

ESCALATION AND CONTINUITY

- A named escalation path on both sides
- Problems raised early, with options
- Incident cover only as agreed, never assumed
- Context kept in runbooks and decision records, so a team change does not reset the work

We work to keep engineers who know your system close to it. Continuity comes from shared records, not from a promise that one person stays forever.

WHAT A MILESTONE LOOKS LIKE

A first-milestone acceptance record

FICTIONAL SAMPLE Invented engagement. Bracketed labels replace every name and identifier. This is not a customer report.

Engagement: [Client A] platform work | Milestone 1: release and recovery path for [Service X] | Period: [Week N]

ACCEPTANCE CRITERION	EVIDENCE SHOWN	RESULT
Deploys go through the agreed pipeline only	Pipeline run [Run ID] deployed to staging, then production after [Client approver] approval	Met
Rollback rehearsed in staging	Rollback completed; service checks passed; runbook updated	Met
Alert for the agreed user journey	Test alert received by [Client on-call contact]	Met
Runbook handed over and walked through	Session held with [Client engineer]; questions logged and answered	Met
Load check at the agreed level	Moved to milestone 2 by [Client product owner] decision	Carried forward

CHANGE REQUESTED

Add [Service Y] to the scope. Impact stated. Awaiting [Client product owner] approval; no work started.

OPEN RISK

Restore from backup not yet tested with production-like data. Owner: [Ampity delivery lead].

DECISIONS NEEDED FROM YOU

Approve or decline the change request. Confirm the milestone 2 acceptance tests.

SIGN-OFF

Accepted with one item carried forward. [Client product owner], [Ampity delivery lead].

WHAT YOU KEEP, AND HOW IT ENDS

You keep the work, and the exit is planned

WHAT YOU KEEP

- Code in your repositories
- Infrastructure in your accounts, defined as code
- Runbooks, decision records and test evidence
- Credentials in your own vault

A PLANNED EXIT

- Handover walkthrough with your team
- Open work and known risks written down
- Access transferred to the people you name
- Outgoing access removed on your instruction

TEMPLATES AND PROOF

[Working with Ampity: FAQ](#)

[Engineering responsibility map](#)

[Sample architecture assessment](#)

[Team Sports Enterprises story](#)

FAQ and map: how scoped work is set up. Sample: fictional.

“The team operated with a genuine sense of ownership, treating our platform as if it were their own.” CEO, sports technology company, Clutch review, 9 October 2025

[Read client reviews on Clutch](#)

Ownership follows the agreed engagement terms; the controls are the ones written into your scope.

Been let down by a vendor before? Start with a scoped first milestone. ampity.io | hello@ampity.io