

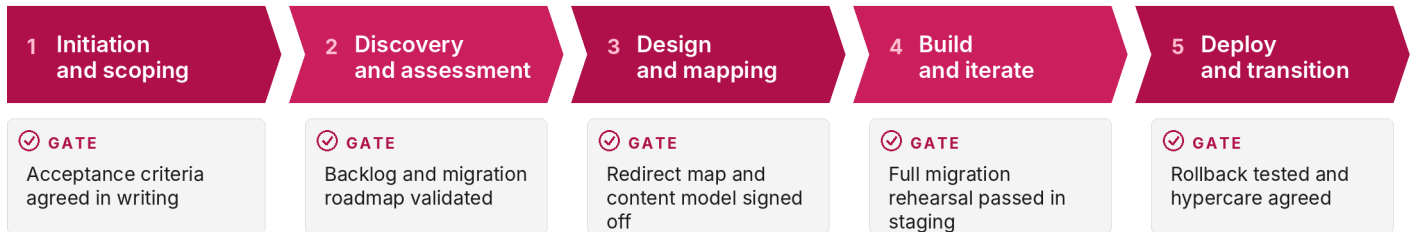


How we run a platform migration that does not slip

Moving a live platform without losing the date, the data or the search rankings

Migrations rarely slip because the build was hard. They slip because the data turned out worse than anyone modelled, because the integration architecture was mapped after the estimate was given, or because governance could not flex once something moved. Our method puts a gate at each of those points. Nothing moves on until its gate is passed, and a gate is evidence rather than an opinion.

THROUGHOUT RAID log open from day one · formal change control on every change · burn rate monitored against baseline



RHYTHM Weekly written progress report · monthly governance review · defined escalation path · one named UK point of contact

	Stage	What we do	Gate that must pass before the next stage starts
1	Initiation and scoping	Confirm objectives, scope, stakeholders, constraints, deliverables and acceptance criteria in writing. Agree the reporting rhythm and escalation path. RAID log opened on day one.	Written acceptance criteria and a baselined plan, agreed before substantive work begins.
2	Discovery and assessment	Review the current estate: applications, integrations, data flows and hosting. Requirements workshops covering functional and non-functional needs. Full pre-migration audit of the source platform: content inventory, URL structure, performance baseline.	A prioritised backlog, a validated migration roadmap, and every integration dependency identified and owned. No build starts on an assumption.
3	Design and mapping	Solution and integration architecture. Content modelled once and mapped to the target. Clean target URL structure with 1:1 redirect mapping. Metadata and structured data mapped. WCAG 2.2 AA embedded from the wireframe stage.	Redirect map and content model signed off. The most common cause of a late slip, closed before build rather than during it.
4	Build and iterate	Short time-boxed sprints with continuous integration and testing, and a show and tell at the end of each. Functional, regression, cross-browser, cross-device, accessibility and load testing throughout. Migration tooling dry run against real data.	A full migration rehearsal in a staging environment that mirrors production, with defects closed to agreed severity thresholds.
5	Deploy and transition	Release through version-controlled pipelines against a deployment validation checklist. Redirects and rankings validated in staging before Go-live. Phased or simultaneous cutover. Documentation, role-based training and knowledge transfer.	A rollback plan tested rather than only written, and a hyper care period with agreed response and fix times before steady state.

Five controls that hold the date

- The RAID log is opened on day one and worked every week, not written once and filed.
- Activity-based estimation against the agreed backlog, burn rate monitored against baseline. Drift becomes visible in weeks, not months.
- Formal change control. Every change carries a transparent cost and schedule impact before it is accepted, so scope cannot arrive quietly.
- Independent review. On larger engagements deliverables are reviewed from outside the delivery team, so quality is never signed off by its author.
- Weekly written progress reporting and a monthly governance review, with a defined escalation path and one named UK point of contact.

What we need from you

- One named decision maker and a single approval chain. Two chains is the most reliable way we know to lose a fortnight.
- Content authoring stays with your team. We provide the structure, the tooling and the quality assurance. You own the words.
- Early access to the source systems, and to the person who actually knows the data rather than the documentation describing it.

Ore Technologies is the single accountable party from first workshop to steady state, whether delivered by our own consultants or by our named partner under our direction.