Delivery roadmap and acceptance pack

SAMPLE - NOT A COMPLETED CLIENT RECORD

Prepared 16 November 2026 | Version 1.0

This reusable example shows how a focused enterprise application release can be scoped, reviewed and handed over. It is generated guidance, not a redacted roadmap, evidence of past delivery, or a signed acceptance document.

Example release boundary

One responsive web application with request submission, approval and status reporting; three permission roles; one documented REST connection. Production-data migration is outside this example. These boundaries illustrate the approved $30,000 application offer; the signed project proposal controls the final scope.

Roadmap driven by acceptance gates

Gate	Reviewable output	Exit condition / owner
1. Scope	Workflow map, role matrix, interface inventory, assumptions and exclusions.	Business owner accepts the first-release boundary and names decision makers.
2. Design	Wireframes, architecture decisions, data ownership and security requirements.	Product and technical owners approve the chosen approach and open dependencies.
3. Build	Working increments, versioned backlog, test records and change log.	Named workflows and interface behaviour are ready for business validation.
4. Acceptance	UAT results, defect register, agreed exceptions and release checklist.	Acceptance owner reviews evidence against the agreed criteria.
5. Release and handover	Deployment/rollback instructions, repository access, operating notes and training record.	Operations owner confirms readiness and the authorised owner approves release.

Planning assumptions

The client supplies an API sandbox, credentials, representative data and available reviewers. Dates and effort are agreed after dependencies are assessed. No universal delivery timeline, response-time SLA or recovery target is implied.

Acceptance checklist

SAMPLE TEST DESIGN - ALL CHECKS NOT EXECUTED

Use this structure to record actual test IDs, environment, build version, expected and actual outcomes, evidence links, defects, reviewer and acceptance date. No pass or sign-off is supplied in this sample.

Check	Expected behaviour to agree	Evidence required / status
AC-01 Roles	A requester can submit; the assigned approver can approve or reject; an administrator manages access. Unauthorised roles cannot perform restricted actions.	Role matrix; positive/negative test records; redacted captures. NOT EXECUTED.
AC-02 Workflow	A submitted request follows the agreed state changes. Invalid or repeated transitions are rejected without corrupting the record.	State-transition tests; audit entries; defect references. NOT EXECUTED.
AC-03 Integration	Valid mapped fields are exchanged. Invalid data and API failures are logged; retries follow the agreed rule and avoid duplicate business records.	Mapping version; request/response logs with secrets removed; failure/retry tests. NOT EXECUTED.
AC-04 Reporting	Status reports respect access permissions and reconcile with the agreed test dataset.	Dataset version; expected/actual counts; role tests. NOT EXECUTED.
AC-05 Quality	Accessibility, load, security and recovery checks meet the requirements explicitly included in the proposal.	Dated reports stating tools, workload/environment, targets and findings. NOT EXECUTED.
AC-06 Release	The selected build can be deployed and rolled back using the operating instructions. Monitoring and incident owners are named.	Rehearsal log; release/rollback record; owner review. NOT EXECUTED.

Acceptance decision

A real acceptance record should state the reviewed release, evidence references, outstanding defects, approved exceptions, decision, authorised reviewer and date. Any untested requirement remains unverified. A template must never be used as proof that a check passed.

Scope and change control

Changes to workflows, roles, interfaces or migration are assessed for cost, schedule and test impact. Record the decision and approval before changing the delivery baseline.

Handover and release readiness

SAMPLE INVENTORY - TRANSFER NOT CONFIRMED

Item	What a completed pack should contain	Owner / verification
Repositories and build	Source repository, release tag, dependency/ licence inventory, build steps and required runtime versions.	Engineering and client technical owner verify access and a clean build.
Configuration and access	Environment inventory, configuration keys, account ownership and secure credential-transfer method.	Client operations/security owners verify access. No secrets in public documentation.
Deployment and rollback	Versioned deployment steps, prerequisites, smoke checks, rollback triggers and recovery instructions.	Release owner records rehearsal and the agreed release decision.
Integration operations	Interface mappings, endpoint owners, scheduling, authentication, rate limits, retries, failure handling and reconciliation.	Both system owners review normal and failed exchanges.
Data and recovery	Data ownership, retention, backup responsibilities, restore procedure and agreed recovery objectives. Migration reconciliation only when in scope.	Data/operations owners attach actual test evidence if commissioned.
User and administrator guidance	Role-based instructions, training agenda, attendance/recording references and known limitations.	Product owner confirms the intended recipients have received the material.
Support and transition	Named contacts, coverage, agreed response targets, escalation and change-request route.	Support terms are separately agreed; no implied 24/7 commitment.

Public evidence: what to share

For a real project, publish only an approved redacted extract. Remove personal data, credentials, private infrastructure, confidential terms and client-sensitive details. Keep enough project context, scope and acceptance structure to make the evidence useful.

How to use this sample

Adapt the example to an agreed project scope, link every check to actual evidence, then obtain review by the named owners. Until that work is done, retain the SAMPLE label and do not describe this document as a client deliverable or completed test report.

Related service: staging.sdlccorp.com/enterprise-software-development-company/