Aegis Security Cloud Modernization

AegisShield - BookEasy Pilot

Fictional technical project management case study

Curated project document pack | 21 documents

This educational case is simulated. AegisShield, BookEasy, the decisions, participants and outcomes are fictional. It does not describe a real client engagement or production deployment.

Contents

01. Sponsor Handoff and Scenario | Before week 1

02. Project Brief | Week 1

03. Project Charter | Week 1

04. Stakeholder Register and Communication Plan | Week 1; updated during delivery

05. Kickoff Meeting Notes | Week 1

06. Implementation Requirements | Weeks 1-2

07. Technical Approach and Delivery Plan | Weeks 1-2

08. Work Breakdown Structure | Week 2 baseline

09. RACI and Team Responsibilities | Week 2

10. Technical Backlog | Week 2 baseline

11. Project Schedule | Week 2 baseline

12. Risk Register | Week 2 baseline; reviewed weekly

13. Test, Migration and Rollback Plan | Weeks 2-4; used through release

14. Project Status Report | End of week 8

15. Issue Log | Week 8; resolved in week 10

16. Schedule Change Request and Decision | Week 9

17. Test Results | Weeks 8-10

18. BookEasy Pilot Review and Acceptance | Week 11

19. Go/No-Go Decision and Release Record | Week 12

20. Operations Handover | Week 13

21. Project Closure and Lessons Learned | Week 13

Sponsor Handoff and Scenario

Timing: Before week 1 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The sponsor hands the PM the proposed pilot boundaries and assessment findings for the BookEasy increment.

Business context

AegisShield is a fictional provider of a cloud security monitoring service (SIEM). Its customer BookEasy runs a scheduling application for businesses. BookEasy sends security events to AegisShield; AegisShield's security analysts investigate the alerts.

Following a leadership change, the CTO commissioned an assessment. The findings and a phased modernization direction are handed to the project team as inputs; the assessment happened before this implementation project.

Assessment handoff

Area	Finding supplied to the project
Runtime and dependencies	Node.js 14 and PostgreSQL 12 are out of community support; Express, Passport and Axios require compatibility and security review.
BookEasy connector	Retry behavior needs stronger duplicate protection, and delivery failures are not sufficiently visible.
Operations	Release, restore and monitoring procedures are incomplete; existing detections and dashboard must continue to work.

Sponsor request

Deliver a controlled first modernization increment for BookEasy. Move selected runtime and database components toward supported versions, improve retry and duplicate handling, make delivery failures visible, and preserve existing login, event ingestion, alerts, and dashboard behavior. Preserve service continuity, protect customer data, and secure approval before production. The initial planning assumption is 12 weeks; capacity and acceptance criteria are confirmed with the delivery team, sponsor, and BookEasy.

Project Brief

Timing: Week 1 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The PM summarizes the business need, target state, and proposed BookEasy pilot scope.

Business need

AegisShield's assessment identified unsupported technology and reliability gaps in the BookEasy connector. The SIEM continues to operate, but these findings increase maintenance and service-delivery risk.

Project objective

Modernize selected SIEM core components and improve the BookEasy connector, then validate the changes with a controlled pilot. The target state preserves existing login, event ingestion, detections, alerts, and dashboard behavior while improving supportability, duplicate protection, and visibility into delivery failures.

Proposed scope

Included	Excluded
Supported runtime and compatible dependencies	A new product or complete SIEM redesign
BookEasy connector retries, deduplication, and monitoring	Modernization of every customer connector
PostgreSQL migration rehearsal, testing, pilot, and handover	Rollout to all customers

Key stakeholders and timing

The CTO sponsors the work. The project manager coordinates Engineering, Database, Security, DevOps/SRE, QA, Customer Success, and BookEasy. The initial planning assumption is 12 weeks; resource capacity, dependencies, and detailed acceptance criteria are confirmed during planning.

Document 03 | Initiation

Project Charter

Timing: Week 1 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The sponsor authorizes the project, its boundaries, responsibilities and success criteria.

Purpose and authorization

The CTO sponsors and authorizes the project manager to coordinate a BookEasy pilot of the SIEM core modernization. The project aims to restore supportability and improve event delivery without rebuilding the product.

Methodology

This project uses a hybrid approach. We use a planned approach for scope, risks, approvals, dates, and release decisions. We use small delivery and test steps for the technical work, so the team can find and fix issues before moving forward.

Scope

Included	Excluded
Supported Node runtime and compatible application dependencies	A new product or complete SIEM redesign
BookEasy connector retries and deduplication	Modernization of every customer connector
PostgreSQL migration rehearsal and controlled release	A wholesale redesign of the dashboard or detection rules
Monitoring, testing, pilot, handover and close-out	Rollout to all customers

Main deliverables

The project produces an approved baseline and delivery plan, current-state and dependency evidence, migration and restore rehearsal evidence, BookEasy reliability and monitoring changes, test and rollback evidence, pilot acceptance, release approval, and operations handover.

Success and constraints

Acceptance requires supported target versions, a successful restore rehearsal, passing agreed login, event ingestion, alert and dashboard tests, and BookEasy validation of the pilot sample. No unresolved critical release-blocking defect may remain.

The team must protect customer data, maintain the existing service during preparation and use an approved change window and rollback plan for production. Budget and team capacity are assumed available for this fictional case; no actual budget is claimed.

Initial risks and resources

Initial risks include regressions after the runtime update, an incomplete database restore, and duplicate events during connector timeouts. The fictional plan assumes Engineering, Database, DevOps, Security, QA, Product, Customer Success, and BookEasy reviewers are available during their assigned work windows. The Project Manager confirms capacity with the Engineering Manager and functional leads during week 1. No specific budget is claimed.

Authority and approval

The project manager coordinates the plan and escalates changes and risks. Technical leads own their designs and validation. The CTO approves scope changes and the production go/no-go decision. In this fictional case, the sponsor approves the charter at the end of week 1 on January 12, 2026.

Stakeholder Register and Communication Plan

Timing: Week 1; updated during delivery | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The PM identifies the people involved and agrees how each group will receive project updates.

Purpose

The PM maintains this fictional register from project initiation through closure. Roles are used instead of names; the register and communication cadence are updated when responsibilities or project needs change.

Stakeholder register

Stakeholder	Interest and role	Key involvement
CTO / project sponsor	Business outcomes, customer continuity and delivery risk	Authorizes the project, significant schedule changes and production go/no-go
Project manager	Delivery coordination, risks and visibility	Maintains the plan, coordinates decisions and escalates blockers
Engineering lead and backend engineer	Technical compatibility and connector reliability	Define the design and implement runtime and connector changes
Database engineer	Data integrity and recoverability	Plans the database migration and validates restoration
DevOps / SRE	Service availability and operational readiness	Owns deployment, monitoring and the operational rollback procedure
Security engineer	Authentication and security controls	Reviews the design, security findings and release evidence
QA engineer	Product behavior and test coverage	Plans and executes tests and validates defect fixes
Customer Success and BookEasy representative	Pilot coordination and customer impact	Coordinate customer communication and validate agreed pilot scenarios

Communication plan

Audience	Information and channel	When
Sponsor	Written status, decisions and escalations	Weekly and at decision gates
Delivery team	Working meeting: progress, dependencies and blockers	Twice weekly
BookEasy and Customer Success	Pilot preparation, expected impact and results	Weekly during pilot preparation and execution
Security, QA and operations	Design, test evidence and release readiness reviews	At agreed review gates

Escalation

The PM alerts the sponsor promptly when a material schedule or scope change needs a decision. Technical owners raise security, data integrity and service-continuity risks as soon as they are identified.

Document 05 | Initiation

Kickoff Meeting Notes

Timing: Week 1 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The project team confirms the objective, responsibilities, first actions and open questions.

Meeting details

Field	Record
Date	January 13, 2026 (simulated)
Purpose	Align the team on the approved implementation project and immediate planning work
Facilitator	Project manager
Attended	CTO, PM, Engineering, Database, DevOps/SRE, Security, QA and Customer Success
Invited	BookEasy representative for pilot planning; attendance to be confirmed
Source	Sponsor handoff, project brief and approved project charter

Discussion and agreements

Topic	Agreement
Objective and scope	Modernize the SIEM core and validate the first increment with BookEasy; other customer rollouts are outside scope.
Schedule	Twelve weeks is the initial planning assumption; team capacity and dependencies require confirmation.
Responsibilities	Engineering leads technical design; Database, DevOps, Security and QA own their specialties; PM coordinates the plan and communication.
Communication and escalation	The team will use the Communication Plan for updates and will escalate material scope, schedule, security, data, and service-continuity risks to the PM and sponsor.
Release authority	Production requires agreed test evidence, operational readiness and a sponsor go/no-go decision.

Open questions

Question	Owner	Next step
Which supported versions are compatible with the service?	Engineering lead	Review dependencies and propose targets in the technical approach
How will the database migration and restore be rehearsed?	Database engineer / DevOps	Prepare an approach and test it in staging
Which events and outcomes will BookEasy use to validate the pilot?	QA / Customer Success	Agree sample scenarios with the customer representative

Actions

Action	Owner	Due	Status
Prepare technical approach and identify compatibility risks	Engineering lead	January 20, 2026	Pending
Draft migration rehearsal and restore approach	Database engineer	January 20, 2026	Pending
Draft test scenarios and pilot acceptance criteria	QA	January 21, 2026	Pending
Record risks, owners and schedule dependencies	Project manager	January 21, 2026	Pending

Implementation Requirements

Timing: Weeks 1-2 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The team agrees what the modernization must accomplish and how each requirement will be checked.

Requirements baseline

Status: Draft for review. The PM coordinates requirements with Engineering, Database, DevOps, Security, QA, Customer Success and BookEasy. Technical leads confirm feasibility, and BookEasy confirms customer validation scenarios. The sponsor approves the requirements baseline at the end of week 2. After approval, the PM records proposed changes and their effects on scope, schedule, resources and risk; the sponsor approves baseline changes before implementation.

Requirements and acceptance

ID	Requirement	Acceptance and evidence
REQ-01	Run on a supported Node.js version with compatible dependencies reviewed for security.	All agreed login, API and regression tests pass; Security confirms no unresolved release-blocking finding. Evidence: version inventory, dependency review and test results.
REQ-02	Move the database to a supported PostgreSQL version while preserving the agreed migration data set.	Migration rehearsal succeeds; record counts and agreed integrity checks show no unexplained missing or altered data; database tests pass. Evidence: rehearsal report and reconciliation results.
REQ-03	Preserve event IDs during retries and prevent duplicate processing within the agreed retry window.	The same event, including a simulated timeout and retry, produces one processed event in each agreed scenario. Evidence: retry results and processing logs.
REQ-04	Preserve login, ingestion, detection, alert and dashboard behavior for agreed BookEasy scenarios.	Approved regression and end-to-end scenarios pass; BookEasy confirms expected outcomes. Evidence: QA results and pilot acceptance record.
REQ-05	Make connector failures and service health visible to operations.	Simulated failures appear in monitoring and notify the agreed operations contact within the agreed response threshold. Evidence: monitoring test results and notification records.
REQ-06	Document and test database restoration and application rollback before production.	Both procedures pass in staging, meet agreed recovery targets, and confirm service and data integrity after recovery. Evidence: procedures and rehearsal results.
REQ-07	Protect customer data during migration, testing and release.	Security verifies access, encryption, approved test data, and logs without credentials or unauthorized sensitive data. Evidence: security review and data handling checks.
REQ-08	Obtain explicit sponsor approval for production deployment.	Release record includes passing evidence, BookEasy acceptance, operational readiness, tested rollback, approved change window, and no critical blocker. Evidence: release checklist and sponsor decision.

Details to confirm before baseline approval

By the end of week 2, leads confirm target versions, migration data boundaries, retry scenarios and window, expected functional results, monitoring thresholds and recipients, and recovery targets. QA records these details in the acceptance checklist.

Boundaries

These requirements apply to the SIEM core changes needed for the BookEasy pilot and the BookEasy connector. They do not authorize a full platform rebuild, dashboard redesign or rollout to other customers.

Technical Approach and Delivery Plan

Timing: Weeks 1-2 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

Technical leads define the target, migration sequence and release safeguards.

Service flow

BookEasy systems > connector > AegisShield event API > PostgreSQL > existing detection rules > analyst dashboard. The connector prepares and delivers events; an HTTP library such as Axios can make requests, but is not itself the connector.

Methodology used

This project uses a hybrid approach. We use a planned approach for scope, risks, approvals, dates, and release decisions. We use small delivery and test steps for the technical work, so the team can find and fix issues before moving forward.

Planned changes

Area	Approach	Owner
Runtime and libraries	Move Node 14 to Node 24 LTS; review Express, Passport and Axios and update compatible versions after testing.	Engineering lead
Database	Rehearse PostgreSQL 12 to 17 using a supported migration method and verify restore and data integrity.	Database engineer
Connector	Use stable event IDs and controlled retries; suppress duplicates and expose delivery failures.	Backend engineer
Operations	Monitor ingestion failures, queue/retry behavior and service health; document on-call procedures.	DevOps/SRE

Delivery safeguards

Build and test in a non-production environment using sanitized or synthetic events. Rehearse the migration and restore, then use a controlled production window with a tested rollback. The team compares versions and compatibility before deploying; target versions are fictional case choices, not advice to upgrade blindly.

Delivery sequence and technical clarifications

The sequence is: confirm compatibility and target versions; implement in a test environment; rehearse migration and recovery; run functional, security, integration, duplicate, monitoring and rollback tests; validate the pilot with BookEasy; then request sponsor approval for production. Proposed targets are Node.js 24 and PostgreSQL 17, pending compatibility validation and technical-owner approval. The connector checks a stable event ID against processed-event records before writing, so retries do not create duplicates. During migration, incoming events follow the agreed controlled path and are tracked for replay or reconciliation. Database restore recovers data; application rollback returns the prior application version. No separate queue is assumed in this case; monitoring tracks request, retry and delivery status directly. The migration rehearsal must pass before production approval.

Work Breakdown Structure

Timing: Week 2 baseline | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The approved project scope is organized into deliverables and manageable work packages.

WBS hierarchy

The WBS describes what the project will deliver. The project is divided into project management, modernized application, database migration, and validation and delivery. The lower-level packages are: 1.1 approved project baseline, 1.2 project reporting and decisions, 1.3 project closure; 2.1 validated runtime and dependencies, 2.2 reliable BookEasy connector, 2.3 operational monitoring; 3.1 validated migration and recovery procedures, 3.2 verified production database migration; 4.1 integrated test results, 4.2 BookEasy pilot acceptance, 4.3 approved application release, and 4.4 operations handover. Detailed activities, owners and acceptance checks are recorded in the backlog and responsibility matrix.

Deliverables and work packages

WBS ID	Deliverable or work package	Completion evidence
1.0	Project management	Coordinated and closed project
1.1	Approved project baseline	Charter, requirements, schedule and responsibilities recorded
1.2	Project reporting and close-out	Status, decisions and closure recorded
2.0	Modernized application	Supported application ready for integration testing
2.1	Runtime and dependencies validated	Selected supported runtime and compatible libraries tested
2.2	BookEasy connector and monitoring validated	Retry, duplicate handling and operational alerts tested
3.0	Database migration	Migrated database and recovery readiness
3.1	Migration and restore rehearsal completed	Reconciliation and restore evidence recorded
3.2	Database migration completed	Approved database migration executed and verified
4.0	Validation and delivery	Customer-accepted pilot and controlled release
4.1	Integrated test evidence completed	Agreed functional, security and recovery checks completed
4.2	BookEasy pilot accepted	BookEasy accepts agreed pilot scenarios
4.3	Application and connector production release authorized	Sponsor go/no-go and controlled deployment recorded
4.4	Operations handover completed	Runbook and support responsibilities transferred

RACI and Team Responsibilities

Timing: Week 2 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The team agrees who carries out, owns, advises on and receives updates about each deliverable.

How to read the matrix

The matrix uses R, A, C and I to show responsibility, accountability, consultation and updates. Roles are fictional and can represent more than one person.

Responsibility matrix

Deliverable or decision	Responsible (R)	Accountable (A)	Consulted (C)	Informed (I)
Project baseline approval	PM	CTO / sponsor	Engineering lead, Product	Delivery team
Project reporting	PM	PM	CTO / sponsor	Delivery team
Runtime and connector changes	Backend engineer	Engineering lead	Security, QA, DevOps	PM
Database migration and restore	Database engineer	Database engineer	DevOps, Engineering, QA	PM, Security
Security and data protection validation	Security engineer	Security lead	Database, DevOps, QA	PM, CTO
Test evidence and defect validation	QA engineer	QA engineer	Engineering, Security, Database	PM
BookEasy pilot acceptance	Customer Success	BookEasy representative	QA, Engineering, PM	CTO
Production go/no-go authorization	PM coordinates evidence	CTO	Engineering, QA, Security, DevOps, BookEasy	Delivery team
Production deployment execution	DevOps / SRE	Engineering lead	Database, QA, Security, PM	CTO, Customer Success
Operational handover acceptance	DevOps / SRE	Support lead	Engineering, Database, QA, PM	Customer Success

Key terms

Responsible: The role that performs the work. Accountable: The role that answers for the result or makes the final decision; each row has one accountable role. Consulted: People whose input is needed before completing the work or making a decision. Informed: People who receive updates but do not need to approve the work. Baseline: An approved version of the project plan used to track progress and assess changes. Go/no-go: A formal decision to proceed with or postpone a release. Handover: The transfer of the service, documentation and support responsibilities to the operations team.

Technical Backlog

Timing: Week 2 baseline | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The work packages become prioritized tasks with owners, dependencies and acceptance evidence.

Prioritized work

ID / WBS	Task / owner	Dependency	Acceptance evidence
B01 / 2.1	Update runtime and review dependencies / Engineering	Approved technical approach	Login, API and regression tests pass on selected versions (REQ-01)
B02 / 2.2	Implement stable IDs, retries and deduplication / Backend	Agreed BookEasy event scenarios	The same test event is processed once after a simulated timeout (REQ-03)
B03 / 3.1	Rehearse migration and restore / Database	Prepared test environment and representative data	Records reconcile and restore succeeds (REQ-02, REQ-06)
B04 / 2.2	Configure ingestion monitoring / DevOps	Connector test environment	A simulated failure generates an actionable alert (REQ-05)
B05 / 4.1	Run integration and security checks / QA	B01-B04 ready for testing	Agreed tests pass with no release-blocking defects (REQ-01-06)
B06 / 4.2	Coordinate BookEasy pilot / Customer Success	B05 and customer test scenarios	BookEasy accepts documented pilot results (REQ-04)

Release and handover work

ID / WBS	Task / owner	Dependency	Acceptance evidence
B07 / 3.2, 4.2	Execute approved migration and release / Database and DevOps	Successful pilot, restore rehearsal and sponsor go/no-go	Production checks and agreed reconciliation completed; rollback available
B08 / 1.2, 4.3	Transfer support procedures and close project / PM and DevOps	Release and stabilization	Operations accepts runbook; sponsor accepts closure

How estimates are used

Effort estimates are initial team estimates for planning, based on dependencies and assumed capacity. Functional leads confirm or change them during planning. The PM records estimate changes and their schedule impact.

Backlog controls and key terms

The backlog uses priority and status to distinguish pending, in-progress, blocked and done work. Monitoring is WBS 2.3. Security and data protection is a separate task under WBS 4.1. Release execution, operations handover and project closure are separate tasks. Backlog: ordered project work. Dependency: work required before another task. Acceptance evidence: proof of completion. Retry: another delivery attempt after temporary failure. Deduplication: preventing the same event from being processed twice. Regression testing: checking existing behavior after a change. Runbook: operating and recovery instructions.

Project Schedule

Timing: Week 2 baseline | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The team maps the initial twelve-week milestones and their dependencies.

Original schedule

Weeks	Milestone
1-2	Charter, design, WBS, ownership and backlog
3-4	Environment, backups and runtime work
5-7	Connector, database migration rehearsal and monitoring
8-9	Integration, security and recovery tests
10	BookEasy pilot
11-12	Release, stabilization and handover

Dependencies and baseline control

Connector and database changes must be ready before end-to-end tests. A restore rehearsal and QA/Security evidence are required before the pilot. Sponsor release approval follows pilot acceptance. The CTO / sponsor approves the schedule baseline at the end of week 2; version 1.0 becomes the reference for tracking and changes. The part-time team availability was reviewed with functional leads when building this estimate. The PM records competing work that could affect the schedule.

Schedule tracking view

Planned work is tracked separately from milestones: planning and design (weeks 1-2), environment and runtime (weeks 3-4), connector, migration rehearsal and monitoring (weeks 5-7), testing (weeks 8-10), BookEasy pilot (week 11), production authorization and deployment (week 12), stabilization and operations handover (weeks 12-13), and project closure (week 13). Key terms: schedule baseline is the approved time plan; milestone is a significant completed result; dependency is work required before another activity; stabilization is the post-deployment monitoring and handover period.

Capacity, critical path and advance conditions

Planning capacity assumptions are: PM 50%, Engineering 50%, Database 25%, DevOps/SRE 25%, Security 15%, QA 50%, Product 10%, Customer Success 20%, and Support 10%. These are planning assumptions, not billing claims; the PM records competing work. The likely critical path is compatibility, test environment, migration and recovery rehearsal, integration and security testing, BookEasy pilot, sponsor authorization, deployment, stabilization, handover and closure. Functional tests must pass before the pilot, BookEasy must accept the pilot before production authorization, stabilization must pass before handover, and handover must be accepted before closure. Calendar duration is the time an item occupies the schedule; effort is the person-time required, so a two-week testing window may use four QA days.

Gantt and resource histogram

The Gantt is the primary schedule view for activities, dependencies and milestones across the 12-week plan plus the one-week recovery impact. A resource histogram compares weekly planning hours with each part-time role's availability; it is used to identify conflicts and competing work. A burndown is not included because this project is not organized into sprints.

Document 12 | Planning

Risk Register

Timing: Week 2 baseline; reviewed weekly | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The team records what could threaten delivery and assigns owners to reduce or manage each risk.

How this register is used

This is the fictional week-2 planning baseline. Probability and impact are qualitative estimates, not measured statistics. The PM reviews open risks with their owners weekly; an event that later happens is recorded separately as an issue in a dated update.

Scoring criteria

Low probability means the event is not expected under the current plan; Medium means there is a credible chance; High means the event is likely without action. Low impact means limited rework; Medium means a material delay or contained customer effect; High means a release blocker, data-integrity concern, service disruption, or major schedule impact.

Initial risks

ID / risk	Probability / impact	Response and trigger	Owner
R01 - Dependency incompatibility	Medium / High	Engineering checks compatibility and runs login and API regression tests in staging; escalate if a required library blocks the selected runtime.	Engineering lead
R02 - Data loss or inconsistent migration	Medium / High	Database engineer validates backups, rehearses migration and restore, and reconciles agreed sample records; stop the release if recovery or reconciliation fails.	Database engineer
R03 - Missing or duplicate events	Medium / High	Backend and QA test stable event IDs, retry timeouts and duplicate handling; block the pilot if agreed test events are missing or duplicated.	Engineering lead
R04 - Customer disruption	Medium / High	DevOps plans monitoring and a controlled change window; Customer Success coordinates notice with BookEasy; pause the release if service health fails the agreed checks.	DevOps / SRE
R05 - Pilot or schedule delay	Medium / Medium	PM tracks dependencies and test readiness weekly; escalate a forecast slip to the sponsor and agree any schedule change.	Project manager

Review and escalation

Risk owners report changes to the PM. If a risk occurs, the team records an issue with impact, actions and status; the PM communicates material changes to the sponsor and affected stakeholders. The baseline does not assume that any listed risk has already occurred. Later updates may record an issue and the date it materialized.

Risk controls and definitions

Each risk is managed through prevention, a trigger, and contingency. Risk: an uncertain event that could affect the project. Issue: a risk that has happened or a current problem. Risk owner: the role accountable for monitoring and responding. Trigger: a sign that the risk may be occurring. Mitigation: action taken to reduce probability or impact. Contingency: action taken if the risk occurs. Preventive controls include compatibility testing, restore rehearsal, capacity checks, and data-protection review.

Risk detail baseline

R01: version differences may cause library incompatibility and delay deployment; prevent with compatibility review, trigger on a failed target-runtime check, and revise the target if needed. R02: incomplete migration preparation may cause missing or inconsistent data; prevent with restore rehearsal, counts, integrity checks and reconciliation, and stop release if checks fail. R03: retry timeouts may create duplicate or missing events; prevent with stable IDs and timeout tests, and pause pilot approval if a repeated event is processed twice. R04: part-time capacity may delay work; prevent with availability checks and re-sequencing. R05: weak data controls may expose customer information; prevent with masked data, access and encryption checks, and stop the activity if a control fails. Each risk has an owner, status, last review date, and preventive-action due date in the working register.

Document 13 | Planning

Test, Migration and Rollback Plan

Timing: Weeks 2-4; used through release | Project: Aegis Security Cloud Modernization - BookEasy Pilot

Technical owners define how the team will verify the changes, rehearse the migration and recover if the release fails.

Scope and responsibilities

QA coordinates test evidence; Engineering validates application and connector changes; the database engineer owns database migration and restore checks; DevOps prepares deployment, monitoring and recovery; Security reviews security evidence. Customer Success coordinates BookEasy pilot acceptance. The CTO makes the production go/no-go decision.

Test coverage

Test	Expected result / requirement	Owner
Runtime, login and API regression	Agreed login and API checks pass on the selected supported versions (REQ-01).	Engineering / QA
BookEasy retries and duplicate events	A repeated test event retains its identifier and is processed once, including a simulated response timeout (REQ-03).	Backend / QA
Events, detection and dashboard	Representative test events appear and produce the expected existing test alert (REQ-04).	QA / BookEasy representative
Monitoring	A simulated delivery failure generates an operational alert (REQ-05).	DevOps / QA
Migration and restore rehearsal	Representative records reconcile and a restore completes successfully in staging (REQ-02, REQ-06).	Database engineer
Security review	Access and relevant security findings are reviewed; no unresolved critical release-blocking defect remains.	Security / QA

Migration sequence

In the initial plan, the team prepares staging and verifies backups in weeks 3-4; rehearses the database migration and restoration in weeks 5-7; and completes integration, security and recovery tests in weeks 8-9. The BookEasy pilot was initially planned for week 10.

For the production change, the database engineer and DevOps confirm the approved window, backup and recovery steps, monitoring and customer communication. They execute the approved migration, reconcile the agreed records and validate application behavior before normal operation resumes.

Release and recovery decisions

The sponsor reviews pilot acceptance, test evidence, unresolved defects, operational readiness and the recovery plan before authorizing production. The team stops the change if release checks fail and follows the agreed recovery procedure.

Returning to the earlier database after customers have written new data requires a specific reconciliation or restoration strategy; switching versions alone is not a complete rollback. The database engineer must document and rehearse that strategy, including the agreed data-loss and downtime limits, before the release is approved.

Environment, migration and rollback controls

Testing uses a non-production environment with synthetic or approved masked BookEasy events. Scenarios include normal events, retries, simulated timeouts, duplicate attempts, alerts, dashboard results, migration records and recovery records. QA records expected and actual results with evidence; Security owns security and data-protection evidence, Database owns migration and restore evidence, DevOps owns monitoring and rollback evidence, and BookEasy confirms pilot scenarios. The migration sequence is: approved window and backup; control and record incoming events; migrate; check schema, counts, integrity and connectivity; resume flow; validate service health and record evidence. Rollback triggers include critical service failure, failed integrity or reconciliation checks, unexpected lost or duplicate events, failed login or event flow, or a critical security or monitoring failure. The CTO / sponsor authorizes rollback; the PM coordinates, DevOps reverts the application, and Database restores or reconciles data. Restore and application rollback are separate actions, and post-migration events are protected through replay or reconciliation.

Key terms

Test gate: a checkpoint that must pass before work advances. Migration rehearsal: a practice database change using representative data. Rollback: returning the application or data path to an approved prior state. Reconciliation: comparing records and events to find and resolve differences.

Plan versus outcome

This page describes the week-2 plan, not the result. In the fictional execution, a duplicate-event defect delays the pilot; the issue and updated schedule are recorded in the execution documents.

Document 14 | Execution

Project Status Report

Timing: End of week 8 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The PM reports progress, the test defect and the forecast impact before the sponsor decides.

Report details

Field	Record
Project	AegisShield Security Cloud Modernization - BookEasy Pilot
Reporting period	Weeks 7-8; example weekly report
Issue date	February 27, 2026 (simulated)
Prepared by	Project manager
Overall status	Amber - integration testing identified a pilot-blocking defect
Forecast	Pilot week 11; production release week 12; stabilization and handover week 13; closure week 13
Decision required	Approve or reject a one-week schedule change after reviewing the recovery proposal

Progress and next steps

Area	Current position
Completed in weeks 3-7	Staging prepared; runtime and dependency changes implemented for testing; migration and restore rehearsed; initial monitoring configured.
In progress	QA is executing integration and recovery tests. Engineering is investigating issue ISS-01, concerning duplicate events after a simulated timeout.
Customer impact	The defect was detected in the test environment before the pilot; no production impact is claimed.
Next steps	Engineering proposes a fix, QA prepares a retest, and the PM presents the schedule impact to the sponsor and informs BookEasy through Customer Success.

Status meaning and recommendation

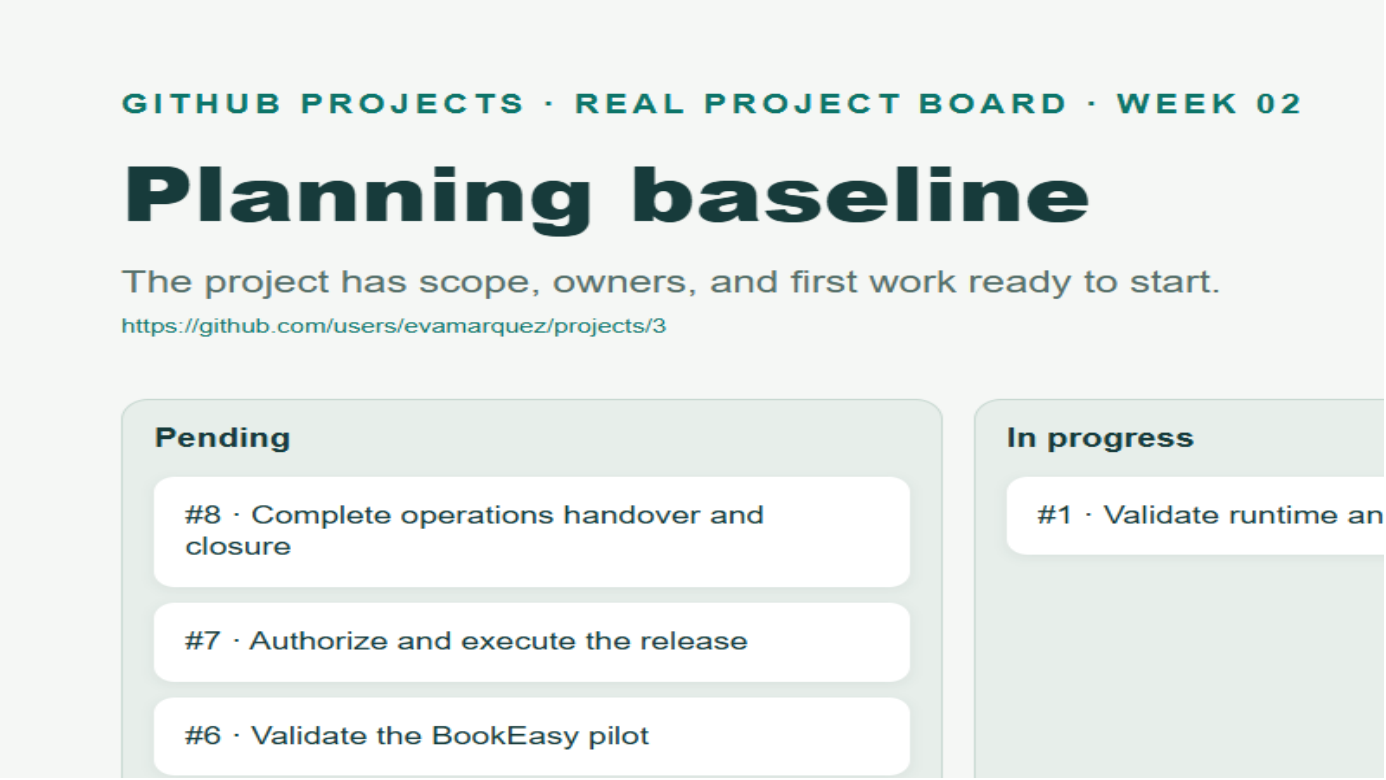
Amber means there is a threat to the schedule that requires action and a decision, but there is still a viable proposal to continue. The PM recommends approving a one-week extension, keeping scope and acceptance criteria unchanged, and completing the fix and retest before the pilot. Sponsor response is requested by March 2, 2026 (simulated) so the team can protect the forecast.

Resources and budget

The delay is currently expected to use existing part-time team capacity. Any additional capacity or budget impact is still under evaluation and will be escalated if the recovery estimate changes.

Execution snapshots

The portfolio uses three GitHub Projects boards at key points: week 2 planning complete, week 8 defect opened and pilot blocked, and week 13 handover complete. Each snapshot is linked to the separate repository and labeled as fictional project evidence.



GITHUB PROJECTS · REAL PROJECT BOARD · WEEK 08

Issue found

ISS-01 is open. The duplicate-event defect blocks the pilot while the t

https://github.com/users/evamarquez/projects/4

Pending

- #8 · Complete operations handover and closure
- #7 · Authorize and execute the release
- #6 · Validate the BookEasy pilot

In progress

- #9 · ISS-01: Duplicate e
- #5 · Run integration, se tests
- #4 · Add delivery and se

GITHUB PROJECTS · REAL PROJECT BOARD · WEEK 13

Project close

The release is complete, the service is stable, and operations has ac

https://github.com/users/evamarquez/projects/5

Pending

In progress

Issue Log

Timing: Week 8; resolved in week 10 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The team tracks the duplicate-event defect, its owner, actions and resolution.

Issue ISS-01

Field	Record
Detected	Week 8 during staging integration tests
Related risk and requirement	R03 - missing or duplicate events; REQ-03 - retain event ID and suppress duplicates in agreed scenarios
Description	When a test request succeeds but the response times out, a retry uses a new identifier and the same event can be processed twice.
Impact	Duplicate stored test events can produce repeated alerts. The BookEasy pilot is blocked until the fix passes retesting.
Owner	Engineering lead; backend engineer implements the correction and QA verifies it
Initial status	Open in week 8; no production incident or customer data loss established

Action and resolution

When	Action / outcome	Owner
Week 8	Reproduce the timeout scenario and trace the identifier change.	Engineering / QA
Weeks 9-10	Preserve the original event ID across retries; review failure monitoring and add timeout and duplicate tests.	Backend / DevOps / QA
Week 10	QA reruns the agreed scenarios; the original test defect no longer reproduces and the issue is closed for the pilot.	QA / Engineering lead

Follow-up

The PM links ISS-01 to the schedule decision, reports progress to the sponsor and coordinates customer communication. The result applies to the agreed test scenarios, not every possible failure condition.

Current status and tracking

Current status: Closed. Closure date: March 13, 2026 (simulated). Priority: High. Target resolution: March 13, 2026. GitHub board reference: github-board-during.md, ISS-01 card. The issue was Open in week 8; that is historical status, not the current status.

Cause, correction and closure evidence

Root cause: when the response timed out, the retry created a new event identifier. Correction: preserve the original event ID and check processed-event records before writing so the same event is processed once. QA retested the timeout, retry, duplicate, and normal-delivery scenarios in week 10. Engineering lead verified the correction, QA recorded the passing results, and the PM accepted the closure evidence against REQ-03. Key terms with case examples: Issue is a current problem that needs action, such as ISS-01 blocking the pilot. Defect is behavior that fails an expected result, such as one event being processed twice. Root cause is why it happened, such as the retry creating a new ID. Retest means repeating the failed scenario after correction. Closure evidence is proof such as passing logs, duplicate checks, and QA results linked to REQ-03.

Schedule Change Request and Decision

Timing: Week 9 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The PM requests a one-week change after ISS-01, and the sponsor records the decision and follow-up actions.

Change request

Requested by: Project Manager. Date requested: February 27, 2026 (simulated). Reason: ISS-01 blocks the BookEasy pilot because a timeout retry can create duplicate events. Proposed change: move the plan from 12 to 13 calendar weeks while keeping scope and acceptance criteria unchanged. The PM recommends delaying the pilot rather than reducing tests. The request uses targeted part-time effort from Engineering, QA, PM and Customer Success; no additional budget is requested at this point. BookEasy and the delivery team confirm availability for the revised dates.

Pre-decision analysis

Area	Impact
Dates	Pilot moves from week 10 to week 11; production from week 11 to week 12; stabilization, handover and closure move to week 13.
Resources	Targeted recovery work is added; a full extra week is not assumed for every role.
Costs	No additional budget is approved; any material impact is escalated.
Risks	Keeping the original date would risk duplicate events and weak acceptance evidence; reducing tests would increase release risk.

Decision record

Field	Record
Reference	DEC-01; related to ISS-01 and risk R03
Requested by	Project manager, based on Engineering and QA's recovery estimate
Date requested	February 27, 2026 (simulated)
Decision maker	CTO / project sponsor
Date approved	March 2, 2026 (simulated)
Decision	Approve a one-week schedule extension; keep the agreed scope and release criteria.
PM recommendation	Delay the pilot rather than reduce testing, because the defect affects event integrity and customer acceptance.
Alternatives considered	Reduce test scope (rejected); keep week 10 without a fix (rejected); add capacity (reviewed but not required).
Decision limit	This schedule decision does not authorize production; a separate go/no-go is still required.

Revised baseline

Milestone	Original baseline	Revised baseline
BookEasy pilot	Week 10	Week 11, subject to passing retest
Controlled production release	Week 11	Week 12, subject to pilot acceptance and go/no-go
Handover and project closure	Week 12	Week 13 after stabilization

Resource and cost impact

The extra calendar week uses targeted part-time Engineering, QA, PM and Customer Success effort for the fix, retest and customer coordination. It does not mean a full extra week for every role. No additional budget is approved in this decision; any material capacity or cost impact must be escalated separately. Engineering, QA and BookEasy confirmed availability for the revised dates.

Schedule version and terms

The approved schedule becomes Revised Baseline v1.1. Forecast means the date the team expects to finish; revised baseline means the sponsor-approved dates used to measure progress. Schedule extension means adding time to the plan. Go/no-go remains the separate decision to proceed with or postpone production.

Post-decision actions

The PM updates the schedule to Revised Baseline v1.1, confirms capacity with Engineering and QA, confirms the revised pilot date with BookEasy through Customer Success, updates the GitHub board snapshot, and communicates the new dates. Engineering and QA report retest results. The revised baseline is not permission to release: the separate production go/no-go remains required.

Test Results

Timing: Weeks 8-10 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

QA records the agreed test outcomes and the retest of the duplicate-event defect.

Test summary

The fictional team ran the agreed checks in staging using synthetic or sanitized events. QA coordinated the evidence with Engineering, Security, Database and DevOps. Results cover the stated scenarios and do not prove that every possible failure has been eliminated.

Results and evidence

Check / requirement	Result	Evidence / owner
Runtime, login and API regression / REQ-01	Pass	QA and Engineering confirmed the agreed login and API tests on the selected runtime.
Database migration and restore / REQ-02, REQ-06	Pass in rehearsal	Database engineer reconciled representative records and restored the test backup in staging.
Retry and duplicate handling / REQ-03	Failed in week 8; passed retest in week 10	ISS-01 recorded repeated test events after a timeout; QA reran the agreed scenarios following the correction.
Event, detection and dashboard / REQ-04	Pass for agreed test scenarios	QA confirmed that test events reached the dashboard and the existing rule produced the expected test alert.
Delivery-failure monitoring / REQ-05	Pass	DevOps confirmed that a simulated delivery failure generated an operational alert.
Security review	No critical release blocker identified	Security reviewed the agreed authentication checks and findings before pilot readiness.

Disposition

ISS-01 was closed for the pilot after the week-10 retest. The database engineer and DevOps also rehearsed the documented recovery and data-reconciliation procedure for a simulated post-cutover write in staging. QA shared the results with the PM, technical owners and Security. Pilot acceptance and the production go/no-go were still separate decisions.

BookEasy Pilot Review and Acceptance

Timing: Week 11 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

BookEasy reviews the agreed pilot scenarios and records its acceptance before release.

Pilot context

Date: March 20, 2026 (simulated). The fictional BookEasy representative reviewed the agreed scenarios in a non-production pilot environment using synthetic or approved masked events. Customer Success coordinated the review after the ISS-01 retest. This pilot is a limited validation before production, not a production deployment or rollout to other customers.

Customer review

Scenario / requirement	Expected result	Observed evidence	Reviewed by
Receive BookEasy events / REQ-04	Events appear in the dashboard and existing alert flow.	Agreed sample events appeared and produced the expected test alert.	BookEasy representative / QA
Retry after timeout / REQ-03	Same event ID is retained and one event is processed.	Repeated event was suppressed rather than processed twice; logs retained.	BookEasy representative / QA
Existing detection and alert / REQ-04	Existing behavior remains available for the agreed scenario.	Test event produced the expected existing alert.	BookEasy representative / Security
Data handling / REQ-07	Approved test data and protected access are used.	Masked or synthetic data and access checks recorded.	Security / BookEasy representative

Acceptance decision

Authorized BookEasy approver: BookEasy representative, Customer Success account owner. Decision: Accepted for the agreed pilot scope on March 20, 2026 (simulated). Conditions: production release still requires the CTO go/no-go, operational readiness, and the approved change window. No unresolved BookEasy-specific observation remains; the acceptance does not establish zero loss or production performance under all conditions.

Key terms

Pilot: a limited validation before production; here, BookEasy validates selected scenarios in a non-production environment. Acceptance: the authorized customer confirms the agreed result is met. Test scenario: a defined situation and expected outcome used for verification. Production release: the separately approved deployment to the live service.

Go/No-Go Decision and Release Record

Timing: Week 12 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The sponsor reviews readiness, authorizes the controlled release, and the team records the execution result.

Readiness review

Gate	Evidence presented	Owner
Application and connector	Agreed regression and retry tests passed; ISS-01 closed for the pilot.	Engineering / QA
Security	Authentication and relevant findings reviewed; no unresolved critical release blocker identified.	Security
Database and recovery	Migration and restore rehearsed; the production recovery and data reconciliation procedure reviewed.	Database engineer / DevOps
Customer	BookEasy accepted the agreed pilot scenarios and received the release communication.	Customer Success
Operations	Change window, service monitoring and recovery ownership confirmed.	DevOps / SRE

Pre-release decision

Field	Record
Decision date	March 23, 2026 (simulated)
Decision	Go
Version	Aegis Security Cloud BookEasy increment v1.1
Change window	March 23, 2026, 10:00-12:00 ET (simulated)
Approver	CTO / sponsor
Conditions	Pilot accepted, tests passed, rollback ready, monitoring active, no critical blocker

Execution record

Field	Result
Executed by	DevOps / SRE and Database engineer
Start and end	March 23, 2026, 10:15-11:20 ET (simulated)
Post-release checks	Login, event ingestion, duplicate protection, alerts, dashboard, monitoring and reconciliation checks passed.
Rollback used	No. Rollback remained available and was not triggered.
Open problems	No critical release problem remained; monitoring continued during stabilization.
Result	Release completed successfully for the approved increment.

Stabilization and handover conditions

Stabilization lasted from March 23 through March 27, 2026 (simulated). DevOps / SRE monitored service health and ingestion failures; Engineering supported defects and Customer Success monitored customer communication. Handover could proceed after post-release checks passed, no critical issue remained, monitoring ownership was confirmed, and recovery procedures were accepted. Any wider customer rollout requires separate authorization.

Key terms

Go/no-go: the decision to proceed or postpone a release. Release: the approved deployment of a version. Change window: the authorized time period for the change. Post-release checks: tests run after deployment to confirm service behavior. Stabilization: the period of monitoring and issue response after release.

Operations Handover

Timing: Week 13 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The delivery team transfers the service, recovery procedures and support responsibilities to operations.

Handover context

Following the fictional week-12 release and stabilization, the PM coordinates the transfer of the BookEasy increment to the teams responsible for ongoing support. These records describe a simulated handover; the case does not claim that real operational files or production systems exist.

Handover checklist

Item	Receiving role	Fictional handover outcome
Service and connector operation	DevOps / SRE	Reviews the supported service configuration and ownership of ingestion monitoring.
Operational alerts and escalation	DevOps / SRE	Receives the agreed process for investigating delivery failures and contacting Engineering or Customer Success.
Database recovery and data reconciliation	Database operations	Reviews the rehearsed restore procedure and the approach for handling writes made after cutover.
Test results and known limitations	Engineering / QA	Receives the agreed test results, ISS-01 resolution and pilot-scope limits.
Pilot outcome and customer communications	Customer Success	Receives BookEasy's fictional acceptance and the communication history.

Delivered documents

Document / version	Delivered by	Received by	Acceptance date
Runbook and recovery plan v1.1	DevOps / Database	DevOps / SRE	March 27, 2026
Test and release evidence v1.0	QA / Security	Support lead	March 27, 2026
BookEasy pilot acceptance v1.0	Customer Success	Support and Product	March 27, 2026
Known limitations and issue log v1.0	PM / Engineering	Support lead	March 27, 2026

Support coverage and escalation

Support coverage is business hours, Monday-Friday, Eastern Time, with the on-call DevOps/SRE contact for critical service or ingestion failures outside those hours. The escalation path is Support lead -> DevOps/SRE -> Engineering lead -> CTO / sponsor for a release, security, data-integrity, or service-continuity decision.

Acceptance and remaining work

On March 27, 2026 (simulated), receiving owners confirmed access to the service, monitoring, logs, runbook, and recovery procedures. They reviewed how to monitor service health and execute restore, reconciliation, and application rollback. Support for this increment moves to the normal operational teams. Assessment and rollout for other customers remain outside this project and require a separate decision.

Open items

Pending item	Owner	Due	Operations acceptance
Confirm first post-handover monitoring review	DevOps / SRE	April 3, 2026	Support lead accepted
Review pilot-scope limitation with Product	Customer Success	April 3, 2026	Support lead accepted

Key terms

Handover: transfer of service, documents and support ownership. Runbook: step-by-step operating and recovery instructions. Escalation: moving an issue to a role with more authority or technical ownership. Known limitations: conditions the service does not cover or that remain outside the project scope.

Project Closure and Lessons Learned

Timing: Week 13 | Project: Aegis Security Cloud Modernization - BookEasy Pilot

The sponsor accepts the project outcome and the PM records results, the delay and lessons learned.

Outcome and acceptance

The fictional project closes in week 13. The original baseline was 12 weeks; the sponsor-approved Revised Baseline v1.1 was 13 weeks after ISS-01. The project completed within the revised approved schedule. BookEasy accepted the pilot in week 11, the sponsor authorized release in week 12, and Operations accepted handover in week 13. No actual customer, production deployment or operational metrics are represented.

Objectives and outcomes

Objective	Fictional outcome / evidence
Modernize core components	The selected supported runtime and database path were tested; the authorized production change and post-change checks were recorded.
Improve BookEasy connector reliability	ISS-01 was corrected and QA confirmed the agreed timeout and duplicate-event scenarios in staging.
Preserve the customer workflow	The BookEasy representative accepted the agreed pilot event, alert and dashboard scenarios; this does not guarantee performance in every condition.
Prepare ongoing support	Operations reviewed monitoring, recovery, responsibilities and known limitations in the week-13 handover.

Schedule and issue

The initial baseline targeted a week-10 pilot and week-12 closure. The duplicate-event defect found in week 8 blocked the pilot. The sponsor approved a one-week shift in week 9; QA verified the correction in week 10, the pilot ran in week 11 and the project closed in week 13.

Final deliverable check

WBS area	Acceptance record	Final status
1.0 Project management	Charter, reporting, decisions and closure records	Complete
2.0 Modernized application	Runtime, connector and monitoring evidence	Complete
3.0 Database migration	Migration, restore and reconciliation evidence	Complete
4.0 Validation and delivery	Tests, pilot acceptance, release and handover records	Complete

Outstanding items and resource release

The first post-handover monitoring review and pilot-scope review remain with DevOps/SRE and Customer Success, due April 3, 2026 (simulated), and are accepted by Support as operational follow-up. Final documents are stored in the GitHub case repository and the published portfolio PDF; the final pack was released on March 27, 2026 (simulated). Project resources are released after closure acceptance, with functional teams retaining normal support ownership.

Cost result

No simulated budget or cost baseline was established for this case, so cost variance is not evaluated. The schedule extension used targeted part-time effort; no additional budget was approved.

Lessons learned

Observation	Improvement for future work	Who applies it
Retry timeout behavior revealed a duplicate-event defect late in integration testing.	Include timeout and stable-ID scenarios early in connector development and refinement.	Engineering and QA
A migration needs recovery and data reconciliation decisions before a release.	Agree and rehearse recovery procedures and data-handling limits before go/no-go.	Database, DevOps and Security
Customer validation depended on agreed sample events and availability.	Schedule the customer review and agree pilot scenarios early.	Customer Success and Product
The team used visible decisions, risk tracking and staged evidence to recover.	Repeat the status, decision-log and evidence-gate practice.	PM and functional leads

Closure decision and approval

The CTO / sponsor approved closure on March 27, 2026 (simulated), referencing the Operations Handover acceptance and Revised Baseline v1.1. In this simulated case, the project closes after the approved deliverables are accepted. Any wider customer rollout remains a separate initiative. Key terms: Project closure is formal end of the project. Lessons learned are improvements captured for future work. Outstanding items are follow-ups transferred after closure. Resource release is the return of project capacity to normal work.