QUALITY ENGINEERING / EXAMPLE 04

ILLUSTRATIVE CONTENT · NOT AN APPROVED PROJECT TEST PLAN

Integrated Test Plan

A practical release-quality plan covering functional and non-functional testing, negative cases, offline resilience, traceability, defect handling and formal release criteria.

PREPARED BY
Carlo Hennsa

DOCUMENT REFERENCE
EX-TP-001

VERSION / DATE
1.0 · 14 October 2026

PROJECT
Aster Field Operations · Fictional

01

TEST PLAN / SECTION 01

Purpose and release decision

What must be proven before acceptance

RELEASE ASSURANCE / ILLUSTRATION

Release only when essential workflows and recovery behaviour meet their approved evidence thresholds.

This test plan covers the fictional **Aster Field Operations Platform**, an offline-capable application with identity, field inspections, API synchronization and a supervisory dashboard. It shows how I would connect requirements, scenarios, observed outcomes and release decisions.

Current sample status: planning baseline. No test execution has occurred and no pass results are claimed.

12
Representative cases

04
Quality gates

00
Executed tests

Type: integrated test plan | **Status: sample / not executed** | **Risk focus: offline recovery**

INTERPRETATION

All systems, thresholds, quantities, owners, dates and test cases below are **fictional examples**. Replace them with approved requirements and measured evidence before using this plan for a real system.

02

TEST PLAN / SECTION 02

Scope and assumptions

Explicit boundaries prevent accidental scope expansion

IN SCOPE

Release candidate RC-01

- Authentication, session renewal and role permissions
- Inspection creation, validation and photo evidence
- Offline capture, queued writes and replay after reconnection
- Supervisor review and read-only dashboard reporting
- API contract, database integrity, audit trail and rollback
- Accessibility, security, performance and operational resilience

OUT OF SCOPE

Deferred capabilities

- Industrial gate actuation or other physical safety controls
- Native iOS / Android applications
- Predictive AI hazard recognition
- Multi-region disaster recovery
- Unapproved external vendor integrations

Exclusions require product-owner acceptance, not an engineering assumption.

Entry assumptions to validate

- User stories and acceptance criteria are versioned and approved.
- A controlled test environment and anonymized test dataset exist.
- Service contracts and identity-provider test credentials are available.
- Offline capability means an explicitly defined network outage, not an undefined level of infrastructure failure.

03

TEST PLAN / SECTION 03

System under test and environments

What is deployed, where, and with which dependencies

Field client

Responsive PWA
IndexedDB / local queue
Offline capture

Application API

ASP.NET Core
AuthZ / validation
Idempotent commands

Primary storage

Relational database
Audit evidence
Transactional writes

Supervisor view

Review queue
Compliance dashboard
Read models

Environment	Purpose	Configuration / rule	Evidence expected
TEST	Automated integration	Disposable database, deterministic seed	Pipeline reports + API traces
STAGING	Release candidate acceptance	Production-like auth, TLS and monitoring	Test run IDs + screenshots + logs
DISCONNECTED	Network partition exercises	Block API route at field client; server stays healthy	Queue snapshots + replay trace
RECOVERY	Backup and restore rehearsal	Fresh isolated server; sanitized snapshot	Restore log + record reconciliation

ENVIRONMENT FIDELITY Record application build SHA, database migration version, browser, device type, clock source, feature flags and deployed configuration with each formal test run.

04

TEST PLAN / SECTION 04

Test strategy and coverage

Layered verification with explicit responsibility

LAYER 01

Automated

Unit tests, validation rules, contract tests, repository queries and deterministic replay logic in CI.

LAYER 02

Integrated

API + database + PWA journeys, authorization, idempotency, queue replay and event audit.

LAYER 03

Operational

Failure drills, access review, load tests, restore tests and stakeholder acceptance.

Coverage expectations

- Every high-priority requirement maps to one or more test cases and expected results.
- Every safety- or security-relevant denial path has a negative test, not only a happy-path test.
- Cross-system behaviour is exercised end-to-end, including an API restart and duplicate message replay.
- Test automation runs on each pull request; a release candidate receives full staging validation.

Test level	Focus	Primary owner	Run cadence
Unit	Business rules, validators, mapping, conflict resolution	Development	Every change
Contract / API	Authorization, schema compatibility, error behaviour	Development + QA	CI + release candidate
End-to-end	Field capture → sync → supervisor approval	QA	Daily / release candidate
Resilience / security	Network partition, restore, audit, role separation	QA + platform + security	Before approval
UAT	Approved user journeys, real operators, sign-off	Business owner	Release gate

05

TEST PLAN / SECTION 05

Representative test scenarios

Traceable cases with an observable expected result

These cases illustrate how to record a meaningful test design. For a real project, each ID should reference an approved requirement and a detailed execution script with inputs and attachments.

Case	Scenario / setup	Expected outcome	Priority	Method
TP-001	**Authentication** Valid user logs in; session renewal follows policy	Permitted route works; token expiry enforced	P1	Auto
TP-002	**Authorization** Field observer calls supervisor approval API	403 response; no unauthorized state change; audit event	P1	Auto
TP-003	**Inspection submission** Complete inspection with required image and valid data	One stored record and linked evidence; confirmation shown	P1	Auto
TP-004	**Input validation** Omit mandatory photo or submit invalid area ID	Clear validation response; no partial record saved	P1	Auto
TP-005	**Offline capture** Disconnect field client, capture two inspection records	Records queued locally, distinguishable, survive reload	P1	Auto
TP-006	**Replay after recovery** Reconnect and replay pending queue	Exactly one logical record per command; queue clears on ACK	P1	Auto
TP-007	**Duplicate events** Replay same command ID three times	Idempotent outcome; no duplicate audit or evidence	P1	Auto
TP-008	**Conflicting updates** Client A and B edit same record while disconnected	Defined policy applied; conflict surfaced or rejected; logged	P1	Manual+Auto
TP-009	**Supervisor review** Approve complete inspection with supervisor account	Status advances once; actor, time and reason traceable	P2	Auto
TP-010	**Data isolation** User from Team A queries Team B inspection	No cross-team access; denial visible in logs	P1	Auto
TP-011	**Recovery drill** Restore staging backup on clean infrastructure	Data and audit records reconcile; recovery evidence retained	P1	Manual
TP-012	**Performance smoke** Simulate 100 virtual users for 15 minutes	API p95 <= 800ms for scoped endpoints, error rate <1%	P2	Auto

COVERAGE CAVEAT

Twelve scenarios demonstrate the format only. They are not sufficient to certify the fictional application and do not represent an executed or complete production test suite.

06

TEST PLAN / SECTION 06

Non-functional verification

Thresholds require measured evidence and approval

Quality attribute	Example target	Verification method	Owner
Responsiveness	Selected API p95 <= 800 ms at example workload	Load test + percentile report	QA / Platform
Availability	No lost accepted writes during one API restart	Controlled restart with in-flight queue	Platform
Offline resilience	Queued records persist across 24 h disconnected test	Device reload + reconnect + reconciliation	QA
Security	No privilege escalation or unauthorized cross-team access	Automated negatives + manual review	Security
Accessibility	Core journeys meet agreed WCAG 2.2 AA criteria	Keyboard and assistive technology review	QA / UX
Restore	Example recovery <= 4 h; no unaccounted accepted events	Cold restore drill + data reconciliation	Platform

TARGETS

The numbers above are **illustrative** and must be replaced by an agreed service-level and risk assessment. A performance test without representative hardware, workload and measurement method is not a release argument.

07

TEST PLAN / SECTION 07

Test data and evidence handling

Reproducible results without leaking real information

TEST DATA DESIGN

Controlled fixtures

- Synthetic users with distinct roles and teams
- Inspections with valid, missing and malformed images
- Records with conflicting versions and unique command IDs
- Two seeded databases for initial and post-restore reconciliation
- Known UTC and local-time boundary cases

EVIDENCE PACKAGE

Per release candidate

- Git commit and deployment artifact digest
- Run identifier, environment fingerprint and timestamps
- Machine-readable test result files and failure screenshots
- Selected API / audit traces with sensitive fields redacted
- Defect references, retest result and owner confirmation

PRIVACY

Use synthetic or properly anonymized data in examples and public artefacts. Do not put credentials, real personnel records or production access tokens in test results.

08

TEST PLAN / SECTION 08

Defect management and exit criteria

A release gate is more than a test-pass percentage

Severity	Example	Expected disposition
S1 / Critical	Unauthorized access or loss of accepted data	Blocks release until verified fixed
S2 / High	Offline replay corrupts or silently duplicates data	Blocks release unless formally approved exception
S3 / Medium	Non-critical workflow is degraded with workaround	Document, prioritise and accept explicitly
S4 / Low	Copy, cosmetic or low-impact consistency issue	May defer with tracked ownership

ILLUSTRATIVE RELEASE EXIT GATE

All conditions required

- 100% of P1 scenarios executed and passed on the release candidate.
- No unresolved S1 or S2 defects; exceptions require documented approval.
- All critical requirement IDs trace to executed tests and accessible evidence.
- Offline replay, authorization denial and restore drills completed successfully.
- Production rollback procedure tested and operational ownership accepted.

Gate A: coverage **Gate B: defects** **Gate C: resilience** **Gate D: owner acceptance**

09

TEST PLAN / SECTION 09

Execution sequence and responsibilities

One bounded plan, with re-test time protected

Example test execution window / 4 weeks

Illustrative; depends on release readiness



Role	Responsibility	Approval / output
Carlo Hennsa (example author)	Test approach, review and quality assessment	Recommends readiness; does not self-approve business policy
Development lead	Automated coverage, code fixes, component traceability	CI evidence
QA lead	Scenario execution, evidence, defect triage	Test completion report
Platform / Security	Recovery, monitoring, access and vulnerability checks	Technical assurance results
Business product owner	Acceptance criteria and operational workflow approval	Final business UAT decision

RELEASE AUTHORITY

The roles above are illustrative. In a real engagement, the author, QA function and product owner must have explicitly agreed decision rights; authorship alone does not confer release authority.

10

TEST PLAN / SECTION 10

Document control and reuse

Versioned examples for a consistent technical library

Field	Value
Document title	Integrated Test Plan · Aster Field Operations Platform
Document reference	EX-TP-001
Author	Carlo Hennsa
Version	1.0 · 14 October 2026
Status	Illustrative only / not executed
Source format	HTML + Markdown; PDF rendered from HTML
Framework	Carlo Technical Document Style v1.1
Approval	None required: this is an example, not an operational plan

REUSABLE PRINCIPLE

I structure test plans so every important requirement has a known risk, a clear test method, an observable result and an explicit release decision. I keep proposed thresholds separate from approved acceptance criteria.