

CS-DOC-0008 · GXP-DESK DOCUMENTATION

Phase-gated Validation.

FIT-only redaction. Effective 2026-06-04.

DOCUMENT ID	VERSION	EFFECTIVE	OWNER
CS-DOC-0008	v1.0	2026-06-04	Customer Success

Public — Documentation · Review cycle: On change

Control block and metadata anchor.

The control block identifies the document, its current revision, the regulated process it supports, and the people accountable for its lifecycle. Every value below is the source of truth for any downstream record, audit trail entry, or signature block.

DOCUMENT ID	CS-DOC-0008
TITLE	Phase-gated Validation
VERSION	v1.0
STATUS	FIT-CLEAN
EFFECTIVE DATE	2026-06-04
REVIEW CYCLE	On change
DOCUMENT OWNER	Customer Success
CLASSIFICATION	Public — Documentation
RELATED RECORDS	—
SUPERSEDES	— (initial release)

Sign-off table, ready for ink or e-signature.

The signatures below confirm review and authorisation of this document. Approvals must be recorded in chronological order. If the document is signed electronically, the e-signature record on the GxP-Desk platform supersedes any handwritten entry on this page and carries the same legal weight under 21 CFR Part 11 and EU GMP Annex 11.

Role	Name	Function	Date	Signature
Author		Validation Lead		
Reviewer		Quality Assurance		
Reviewer		Process / System Owner		
Approver		Head of Quality		
Approver		Regulatory Affairs		

What's in this document.

01 — Document Control	2
02 — Approvals	3
03 — Contents	4
01 — What this version covers	5
02 — What this version does NOT cover	6
03 — Phase model	7
04 — PhaseDocumentConfig — phase documents per tenant	9
05 — PhaseDeliverable — deliverable tracking	10
06 — Server Actions	11
07 — Code references	12
Revision History	13
Glossary & Abbreviations	14

What this version covers.

This documentation describes the phase model in the GxP-Desk code:

- **ValidationPhase:** Data model with fields for status tracking
- **Phase status values:** NOT_STARTED, IN_PROGRESS, COMPLETED, BLOCKED (strings)
- **Gate Review:** gateReviewRequired (Boolean) and gateReviewCompletedAt (DateTime)
- **PhaseDocumentConfig:** Per-tenant configuration for document types
- **PhaseDeliverable:** Deliverable tracking per phase
- **Server Actions:** Implemented functions in `app/actions/change/validation-phases.ts`

What this version does **NOT** cover.

This version does NOT describe the following functions envisaged in the original, because they are not implemented in the code:

- **Default 3-phase model (Plan/Execute/Report)** – The code initializes an 8-phase model (Planning, Specification, Build & Configuration, Testing-IQ/OQ/PQ, Reporting, Release), not a fixed set of three
- **Configurable spec gate criteria** – Risk-coverage blocking, inspection-tag blocking, and mandatory RTM coverage are not implemented
- **Two-person QA approval per gate** – The gate signature flow is not coded
- **Optional Specify→Configure gate** – No support for this gate split
- **Gate-removal protection (narrowing-only)** – No validation for gate protection
- **Gate signature as ElectronicSignature** – No GATE_APPROVED meaning integration

Phase model.

ValidationPhase — data structure

The core table `ValidationPhase` (Prisma schema) stores:

Field	Type	Description
<code>id</code>	String UUID	Unique phase ID
<code>changeId</code>	String UUID (FK)	Association with a change
<code>name</code>	String	Phase name (e.g. "Planning", "Specification")
<code>order</code>	Int	Order (1–8)
<code>status</code>	String	NOT_STARTED | IN_PROGRESS | COMPLETED | BLOCKED
<code>startedAt</code>	DateTime?	Timestamp of phase start
<code>completedAt</code>	DateTime?	Timestamp of phase end
<code>completedById</code>	String UUID?	User FK for completion confirmation
<code>gateReviewRequired</code>	Boolean	Default: true. A gate review is required.
<code>gateReviewCompletedAt</code>	DateTime?	When was the gate review performed?

Status values

The values are defined in the code as strings (lines 70–71, `validation-phases.ts`):

```
validStatuses = ['NOT_STARTED', 'IN_PROGRESS', 'COMPLETED', 'BLOCKED']
```

Semantics:

- **NOT_STARTED** – Phase not yet begun
- **IN_PROGRESS** – Phase in progress; `startedAt` is set on transition
- **COMPLETED** – Phase completed; `completedAt` and `completedById` are set
- **BLOCKED** – Phase blocked (e.g. due to open deviations)

Gate Review

The field `gateReviewRequired` (default: `true`) marks whether this phase requires a gate review. The function `completePhaseGateReview(phaseId)` sets `gateReviewCompletedAt` to the current time.

Code reference (lines 86–96, `validation-phases.ts`):

```
export async function completePhaseGateReview(phaseId: string) {
  const db = await getTenantScopedDb();
  return db.validationPhase.update({
    where: { id: phaseId },
    data: { gateReviewCompletedAt: new Date() },
  });
}
```

Default phases on initialization

When a change is created, 8 phases are initialized with pre-configured names and deliverable lists (lines 6–15, `validation-phases.ts`):

- 01 Planning – Validation Plan, Risk Assessment
- 02 Specification – URS, Functional Specification, Design Specification
- 03 Build & Configuration – Configuration Documentation, Data Migration Plan
- 04 Testing - IQ – IQ Protocol, IQ Execution Report
- 05 Testing - OQ – OQ Protocol, OQ Execution Report
- 06 Testing - PQ – PQ Protocol, PQ Execution Report
- 07 Reporting – Validation Summary Report, Traceability Matrix
- 08 Release – Release Approval, Training Completion

Note: This list is hard-coded in the code and cannot be configured per tenant.

PhaseDocumentConfig — phase documents **per tenant.**

The table `PhaseDocumentConfig` (Prisma schema, lines 1152–1173) defines which documents belong to which phase:

Field	Type	Description
<code>id</code>	String UUID	Config ID
<code>tenantId</code>	String UUID?	Tenant association
<code>phaseName</code>	String	Phase name (e.g. "Planning & Requirements")
<code>documentType</code>	String	Document type (e.g. "Validation Plan")
<code>documentTitle</code>	String	Default title
<code>isRequired</code>	Boolean	Is this document required?
<code>isLivingDocument</code>	Boolean	Is this a living document?
<code>templateRequired</code>	Boolean	Is a template required?
<code>templateId</code>	String UUID?	Reference to a template

Usage: Document configurations can be defined per phase for each tenant. There is no configurable gating mechanism based on this config.

PhaseDeliverable — deliverable tracking.

The table `PhaseDeliverable` (Prisma schema, lines 872–886) tracks deliverables within a phase:

Field	Type	Description
<code>id</code>	String UUID	Deliverable ID
<code>phaseId</code>	String UUID (FK)	Phase association
<code>title</code>	String	Deliverable name
<code>description</code>	String?	Optional: description
<code>status</code>	String	PENDING | IN_PROGRESS | COMPLETED
<code>assigneeId</code>	String UUID?	Responsible person
<code>dueDate</code>	DateTime?	Deadline
<code>completedAt</code>	DateTime?	Completion time

Each phase is initialized with a predefined list of deliverables (lines 33–35, `validation-phases.ts`). The status of a deliverable is managed manually.

Server Actions.

initializeChangePhases(changedId)

Initializes all 8 phases with their pre-configured deliverables for a newly created change.

Input: Change ID **Output:** Array of fully initialized ValidationPhase objects **Error case:** Throws an error if phases already exist

getChangePhases(changedId)

Retrieves all phases of a change with their deliverables, assignees, and completion confirmers.

updatePhaseStatus(phaseId, status)

Updates the status of a phase. Automatically sets:

- `startedAt` to now, when `status === 'IN_PROGRESS'`
- `completedAt` to now and `completedById` to the current user ID, when `status === 'COMPLETED'`

completePhaseGateReview(phaseId)

Marks this phase's gate review as completed by setting `gateReviewCompletedAt`.

addPhaseDeliverable(params: { phaseId, title })

Adds a new deliverable to an existing phase (parameters from line 98 onward, not fully read, but present in the code).

Code references.

Primary files:

- `app/actions/change/validation-phases.ts` – Server actions for phase management
- `prisma/schema.prisma` (lines 848–870) – ValidationPhase model
- `prisma/schema.prisma` (lines 872–886) – PhaseDeliverable model
- `prisma/schema.prisma` (lines 1152–1173) – PhaseDocumentConfig model
- `components/system/VModelDiagram.tsx` – V-model visualization (mentioned in the original, not fully reviewed)

Test coverage:

- `app/actions/change/__tests__/validation-phases.test.ts`
- `components/change/__tests__/ChangeDocumentsView.test.tsx`

Line count: 307 lines **Date:** 2026-06-04

REVISION HISTORY

Every change, tracked and signed.

Add one row for every controlled revision. Minor changes (typos, formatting) increment the patch version; substantive edits trigger a fresh review cycle and a new approver round.

Version	Date	Author	Summary of Change	Approver
1.0	2026-06-04	Documentation Team	FIT-only redaction limited to codebase-verified functionality.	Head of Documentation
—	—	—	Reserved for next revision. Do not delete this row.	—

GLOSSARY

Shared language, no ambiguity.

Definitions used throughout this document. Where a term has a specific meaning inside GxP-Desk, the platform-specific definition takes precedence over the generic regulatory term.

CSV	Computerized Systems Validation
GAMP 5	Good Automated Manufacturing Practice, Edition 5 (2nd edition, 2022)
GxP	Good 'x' Practice — covers GMP, GLP, GCP, GDP, GVP
IQ / OQ / PQ	Installation / Operational / Performance Qualification
Part 11	21 CFR Part 11 — US FDA rule on electronic records and electronic signatures
Annex 11	EU GMP Annex 11 — EU rule on computerised systems
URS	User Requirements Specification
FRS	Functional Requirements Specification
RTM	Requirements Traceability Matrix
SOP	Standard Operating Procedure
ALCOA+	Attributable, Legible, Contemporaneous, Original, Accurate (+ Complete, Consistent, Enduring, Available)
ICH Q9	International Council for Harmonisation Quality Risk Management guideline

— End of document —