ComplianceSuite

FIT-CLEAN

CS-DOC-0008 · COMPLIANCESUITE DOCUMENTATION

Phase-gated Validation.

FIT-only redaction. Effective 2026-04-28.

DOCUMENT ID	VERSION	EFFECTIVE	OWNER
CS-DOC-0008	v1.0	2026-04-28	Customer Success

Public — Documentation · Review cycle: On change

01 — DOCUMENT CONTROL

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	2026-04-28
REVIEW CYCLE	On change
DOCUMENT OWNER	Customer Success
CLASSIFICATION	Public — Documentation
RELATED RECORDS	Generated from /content/doc_phasegated.py
SUPERSEDES	— (initial release)

02 — APPROVALS

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 ComplianceSuite 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		

03 — CONTENTS

What's in this document.

01 — Document Control —
02 — Approvals —
03 — Contents —
01 — What this edition covers —
02 — What this edition does NOT cover —
03 — Phase Model —
04 — PhaseDocumentConfig — Phase Documents per Tenant —
05 — PhaseDeliverable — Deliverable Tracking —
06 — Server Actions —
07 — Code Reference —
Revision History —
Glossary & Abbreviations —

01 — WHAT THIS EDITION COVERS

What this edition covers.

This documentation describes the Phase Model in the ComplianceSuite codebase:

- **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 edition does NOT cover.

This edition does NOT describe the following functions intended in the original spec, 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 3-set
- **Configurable Spec Gate Criteria** – Risk-Coverage Blocking, Inspection-Tag Blocking, RTM-Coverage obligation are not implemented
- **Two-Person QA Approval per Gate** – 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
id	String UUID	Unique Phase ID
changeId	String UUID (FK)	Assignment to Change
name	String	Phase name (e.g., "Planning", "Specification")
order	Int	Sequence (1–8)
status	String	NOT_STARTED	IN_PROGRESS	COMPLETED	BLOCKED
startedAt	DateTime?	Timestamp Phase start
completedAt	DateTime?	Timestamp Phase end
completedById	String UUID?	User FK for completion confirmation
gateReviewRequired	Boolean	Default: true. Gate Review is required.
gateReviewCompletedAt	DateTime?	When was Gate Review performed?

Status Values

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

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

Semantics:

- **NOT_STARTED** – Phase not yet started
- **IN_PROGRESS** – Phase is running; `startedAt` is set on transition
- **COMPLETED** – Phase complete; `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 (Line 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 creating a Change, 8 Phases with pre-configured Names and Deliverable list are initialized (Line 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 hardcoded in the code and cannot be configured per Tenant.

PhaseDocumentConfig — Phase Documents per Tenant.

The table `PhaseDocumentConfig` (Prisma Schema, Line 1152–1173) defines which documents belong to which Phase:

Field	Type	Description
id	String UUID	Config ID
tenantId	String UUID?	Tenant assignment
phaseName	String	Phase Name (e.g., "Planning & Requirements")
documentType	String	Document Type (e.g., "Validation Plan")
documentTitle	String	Standard title
isRequired	Boolean	Is this document required?
isLivingDocument	Boolean	Is this a Living Document?
templateRequired	Boolean	Is a Template required?
templateId	String UUID?	Reference to Template

Usage: Per Tenant, document configurations per Phase can be defined. No configurable Gating mechanism exists based on this Config.

PhaseDeliverable — Deliverable Tracking.

The table `PhaseDeliverable` (Prisma Schema, Line 872–886) tracks Deliverables within a Phase:

Field	Type	Description
id	String UUID	Deliverable ID
phaseId	String UUID (FK)	Phase assignment
title	String	Deliverable Name
description	String?	Optional: Description
status	String	PENDING	IN_PROGRESS	COMPLETED
assigneeId	String UUID?	Responsible person
dueDate	DateTime?	Deadline
completedAt	DateTime?	Completion time

Each Phase is initialized with a predefined list of Deliverables (Line 33–35, `validation-phases.ts`). The status of a Deliverable is managed manually.

Server Actions.

initializeChangePhases(changeId)

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 Error if Phases already exist

getChangePhases(changeId)

Fetch 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, if `status === 'IN_PROGRESS'`
- `completedAt` to Now and `completedById` to current User ID, if `status === 'COMPLETED'`

completePhaseGateReview(phaseId)

Marks the Gate Review of this Phase as complete by setting `gateReviewCompletedAt`.

addPhaseDeliverable(params: { phaseId, title })

Adds a new Deliverable to an existing Phase (parameters from Line 98, not fully read, but present in code).

Code Reference.

Primary Files:

- `app/actions/change/validation-phases.ts` – Server Actions for Phase Management
- `prisma/schema.prisma` (Line 848–870) – ValidationPhase Model
- `prisma/schema.prisma` (Line 872–886) – PhaseDeliverable Model
- `prisma/schema.prisma` (Line 1152–1173) – PhaseDocumentConfig Model
- `components/system/VModelDiagram.tsx` – V-Model Visualization (mentioned in Original, not fully verified)

Test Coverage:

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

Line count: 307 lines **Date:** 2026-04-28

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	2026-04-28	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 ComplianceSuite, 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 —