



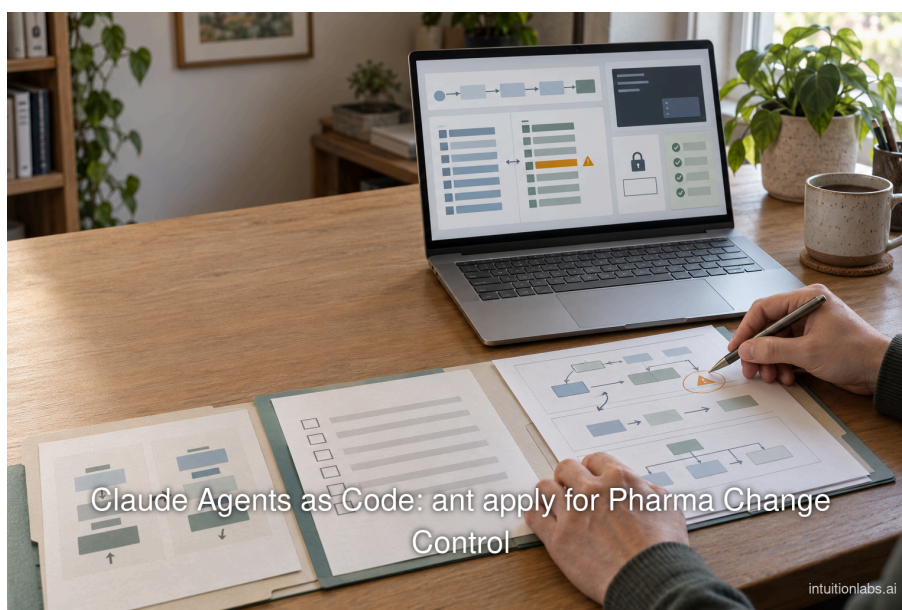
INTUITIONLABS / RESEARCH & PRACTICAL GUIDANCE

Claude Agents as Code: ant apply for Pharma Change Control

Build practical AI for pharma and biotech with IntuitionLabs. We help life-science teams turn complex information and workflows into useful software, governed knowledge systems and AI tools.

Published by IntuitionLabs · 2026-10-13 · claude agents as code · ant apply · anthropic ant cli · claude managed agents · gxp change control · ai agent validation · pharma devops · agent deployment governance · claude lockfile

Original article: <https://intuitionlabs.ai/articles/claude-agents-ant-apply-pharma-change-control>



In this article

1. Executive Summary
 2. Introduction and Background
 3. Key Changes
 4. Implementation Considerations and Process Changes
 5. Drift, Destructive Flags, and Partial Failure
 6. Evidence Packaging and Regression Design
 7. Pilot Design and Operating Decision
 8. Data Analysis and Evidence
 9. Implications and Future Directions
 10. Conclusion
-

Executive Summary

Claude agents as code now has a concrete deployment command: **ant apply**. Anthropic introduced it on **September 3, 2026**, in CLI **1.30.0**, to create and update agents, environments, skills, memory stores, and deployments from repository files. ([platform.claude.com](#)) ([platform.claude.com](#)) For pharmaceutical teams, its value is reviewable configuration promotion. It does not establish that an agent is validated for its intended use. FDA recommends a documented risk assessment for computerized-system validation decisions; EU Good Manufacturing Practice (GMP) Annex 11 calls for controlled configuration changes under a defined procedure. ([www.fda.gov](#)) ([health.ec.europa.eu](#))

The current workflow requires separate treatment of **source files**, **remote resources**, and **claude-lock.json**. Implementation comments distinguish the desired-state fingerprint from the resource version recorded at the last apply. ([github.com](#)) A pull-request dry run is particularly easy to misinterpret: it exits **0 even when its plan is blocked**, so command success cannot be the approval gate. ([platform.claude.com](#)) Proposed controls should evaluate the plan, preserve independent approval, bind production credentials to the intended environment, and test the released agent's behavior. NIST recommends testing before deployment and regularly during operation, including documented test sets and metrics. ([airc.nist.gov](#)) ([airc.nist.gov](#))

State retention matters during failures. A proposed release gate should withhold acceptance until observed outcomes match the approved change and its objectives. ICH Q10 recommends postimplementation evaluation. ([database.ich.org](#)) Pharmaceutical teams should capture the resulting state, reconcile remote outcomes, and authorize completion or restoration explicitly. **The lockfile is neither a validated audit trail nor a rollback system.** Part 11 describes independently recorded, secure, time-stamped operator actions, while PIC/S explicitly distinguishes an audit trail from a change-control system. ([www.ecfr.gov](#)) ([picscheme.org](#))

The implementation blueprint below proposes environment-specific state, externally serialized deployments, narrowly scoped workload identity, controlled exceptions for destructive flags, and an evidence package connecting approval to observed results. Controlled configuration procedures provide the governance frame; SLSA describes provenance linking artifacts to their production process. ([health.ec.europa.eu](#)) ([slsa.dev](#))

These are components to assess within the pharmaceutical quality system, not compliance certifications. The decisive pilot question is whether the organization can reconstruct what was approved, what actually changed, which tests passed, and how dependent runtime state will be recovered.

Introduction and Background

This report examines **Claude agents as code ant apply** for pharmaceutical and biotechnology platform engineering, Development and Operations (DevOps), Machine Learning Operations (MLOps), quality assurance (QA), validation, and system owners. The decision is whether declarative agent configuration can pass through controlled continuous integration and continuous delivery (CI/CD), with enough evidence to support the organization's intended use. It should start with record and process scope: FDA recommends determining whether particular electronic records fall under Part 11 by reference to applicable predicate rules. (www.fda.gov)

Claude Managed Agents provides managed infrastructure for state, memory, permissions, and scheduled execution. (claude.com) Its current overview still describes the service as **beta**. (platform.claude.com) The deployment workflow considered here manages platform resources. A team's acceptance criteria must also cover the business workflow those resources enable. Potential scope questions include whether a manufacturing-support agent drafts a response, retrieves approved material, or initiates a controlled action. Those are proposed scoping questions, not vendor guarantees.

GxP means the family of applicable good-practice requirements. The Medicines and Healthcare products Regulatory Agency (MHRA) describes data-governance expectations spanning laboratory, clinical, manufacturing, distribution, and pharmacovigilance practices. (www.gov.uk) The report does not assume every AI use is a regulated record system. It recommends documenting the intended use, authoritative record location, human review boundary, and consequence of an incorrect output before selecting deployment controls. That approach follows **risk-proportionate quality management** rather than treating every configuration edit identically. (www.ema.europa.eu)

IntuitionLabs' first-party description places identity, authoritative information, citations, and evaluation within its AI implementation approach. (intuitionlabs.ai) The relevant consultancy perspective is therefore integration and accountable operation. The proposed controls remain decisions for the pharmaceutical system owner and quality function.

Release maturity and intended-use risk should guide pilot readiness. WHO likewise describes its health-AI regulatory overview as a high-level resource rather than a regulatory framework or policy. (www.who.int)

Key Changes

Repository declarations become managed resource changes

The launch release named **five resource kinds**; the current guide also supports vaults, requiring CLI **1.34.0 or later** for vault files. (platform.claude.com) (platform.claude.com) The expanded scope means a pilot specification should name the CLI version actually qualified, rather than use the launch minimum as a perpetual compatibility target.

Non-skill resources accept YAML, JSON, or Markdown; skills are directories bundled for upload.

(platform.claude.com) A useful proposed convention is to reserve a directory for each approved kind and require explicit review when a file changes kind. File movement and naming should be assessed as potential resource changes, not dismissed as repository housekeeping.

Table 1 separates the declarative surface from the release evidence that the team should retain. The evidence column is an analyst proposal. Lock entries contain kind, id, optional version, hash, optional remote_hash, and revision or subpath for URL-sourced content. Friendly names do not replace that remote identity. (github.com)

Resource	Source and remote relationship	Version or hash consideration	Proposed evidence
Agent	Instruction and configuration acceptance boundary. (health.ec.europa.eu)	Record remote ID and any version returned.	Approved instruction diff, permissions review, intended-use regression results.
Environment	Infrastructure acceptance boundary. (health.ec.europa.eu)	Compare configuration alongside linked resources.	Network, package, and tool-access acceptance evidence.
Skill	Bundle whose fingerprint covers paths and content. (github.com)	Preserve resolved external dependency revision.	File inventory, dependency review, behavioral tests.
Memory store	Reconciliation manages the container, not runtime memories. (github.com)	Container state does not identify every stored memory.	Separate memory-content, access, retention, and recovery evidence.
Deployment	Markdown body becomes the initiating user message. (github.com)	Evaluate its linked agent and operational settings.	Trigger, input, routing, and run-acceptance evidence.
Vault	File-backed configuration has no prose field. (github.com)	Configuration evidence must be separated from secret material.	Authorized provisioning and recovery procedure, with secrets excluded.

The matrix prevents a common evidence shortcut: proving that configuration was applied does not prove that all data used by the agent was reviewed. The proposed procedure should preserve a separate content boundary, using risk-proportionate assessment consistent with ICH Q9. (www.ema.europa.eu) The appropriate validation scope is the application and its dependencies; Annex 11 separately identifies application validation and infrastructure qualification. (health.ec.europa.eu)

Dependency changes can expand the release boundary

Resources are ordered after the resources they reference. (github.com) This makes the deployment set a dependency graph, not merely a list of edited files. A proposed impact review should identify both direct edits and downstream resources whose effective configuration changes.

The official CLI README states that references record specific versions. (github.com) A proposed external-skill policy should preserve the resolved dependency revision and require review before changing it, using SLSA's resolved-input model as an engineering reference. (slsa.dev) For a regulated release, the review packet should therefore show the resolved dependency set. Reviewers need to understand whether an instruction edit also republishes a coordinator or changes a deployment's effective agent.

Proposed dependency-review questions include:

- **Direct change:** Which resource files changed, and why?
- **Transitive impact:** Which consumers receive changed references or behavior?
- **External content:** Which exact skill revision is included?
- **Unmanaged reference:** Who owns resources referenced by remote ID?
- **Acceptance boundary:** Which dependent workflows require regression testing?

The questions implement the general ICH Q10 recommendation to evaluate proposed changes through quality risk management. (database.ich.org) They also connect to SLSA's distinction between declared build inputs and the dependencies actually resolved during production. (slsa.dev)

The lockfile connects identity and reconciliation

The lockfile is deployment state. Its implementation distinguishes **hash**, the desired state last sent, from **remote_hash**, the normalized remote object after application. (github.com) Normalization excludes server-owned fields and empty values, so a fingerprint should not be described as a comprehensive transcript of every remote field. (github.com)

A proposed release record should retain the before-and-after lockfiles alongside source revision and execution results. A reviewer should be able to connect a source path to the remote resource identity and environment, then examine the behavior that was accepted. The lockfile supplies part of this mapping; approval, operator attribution, electronic-record retention, and signature meaning require their own systems and procedures.

For applicable closed electronic-record systems, **21 CFR 11.10(e)** describes secure, computer-generated, time-stamped audit trails. (www.ecfr.gov) PIC/S also warns against confusing an audit trail with change control. (picscheme.org) Neither a Git commit nor matching hashes resolves those broader questions by itself.

Implementation Considerations and Process Changes

Make pull-request planning an explicit gate

A proposed pull-request workflow should produce a detailed plan from the candidate source set and the correct environment's state. **ant apply --dry-run** . is informational, and its zero exit status includes blocked plans. (platform.claude.com) The CI wrapper should consequently classify the plan outcome and fail the promotion gate when the plan is blocked, its target is unexpected, or its resource changes fall outside the approved request.

Suggested review stages are:

- **Scope:** Link the change request, intended use, and affected records.
- **Integrity:** Identify source revision, CLI build, and resolved dependencies.
- **Plan:** Retain the full resource diff and any blocking messages.
- **Impact:** Map altered resources to applicable regression tests.
- **Approval:** Record independent technical and quality decisions as required by procedure.
- **Release:** Permit mutation only after all required decisions are complete.

These are proposed stages. Annex 11's underlying expectation is controlled change under a defined procedure, not this particular CI design. (health.ec.europa.eu) NIST's generative-AI profile recommends sharing predeployment test results with relevant actors, including release approvers. (nvlpubs.nist.gov)

Where GitHub environments are used, required reviewers can approve deployment jobs and self-review can be prevented. (docs.github.com) Availability must be checked for the actual repository and subscription. An approval screen should identify the revision and plan being approved. Do not let an approver authorize a moving branch without a documented mechanism tying approval to the eventual release input.

After merge, the proposed release job should check out the approved revision, select the production state, and regenerate the plan immediately before mutation. Hold execution if that plan changes the authorized scope or target. Apply the project directory with **ant apply --yes .**, rather than relying on tracked-file reconciliation to discover additions. (platform.claude.com) Preserve the resulting state and execution output whether the job succeeds or fails.

Bind execution to the intended environment

A proposed CI design should use **Workload Identity Federation (WIF)** rather than a stored API key where the trust policy permits. Federation exchanges a trusted identity-provider token for a short-lived Anthropic token bound to a service account. (platform.claude.com) A proposed trust policy should constrain the repository, workflow, environment, and approved branch or revision to the extent supported by the identity provider.

OpenID Connect (OIDC) is the identity layer commonly used for this federation. The receiving provider must enforce trust conditions when exchanging the GitHub-issued OIDC token for an access token.

(docs.github.com) Here, Anthropic requires all configured federation match conditions to pass before accepting the JWT for exchange. ([Anthropic Workload Identity Federation](https://anthropic.com/workload-identity-federation)) Verify the claims actually issued for the repository instead of copying a generic subject string. GitHub's documented subject format changed for repositories created after **July 15, 2026**, introducing immutable owner and repository IDs. (docs.github.com)

Identity evidence should include the account and workspace selected, the trust-rule revision, and the workflow identity. Anthropic's federation documentation notes that a leftover API key can shadow federation credentials. (platform.claude.com) A proposed preflight should reject unintended credential paths and avoid logging secret values.

The implementation also checks the API origin against recorded host, organization, and workspace where values are known. (github.com) That check supports targeting discipline; the release procedure should additionally verify intended operational access. **21 CFR 211.68(b)** requires relevant record changes to be instituted only by authorized personnel. (www.ecfr.gov)

Separate development, test, and production state

Use a proposed layout with shared reviewed templates but separately maintained effective source sets, credentials, and lockfiles for development, test, and production. Select the intended lockfile explicitly and verify its identity before planning. Do not reuse a development lockfile as a convenient way to initialize production: remote identity belongs to the environment already recorded.

The promotion relationship should be:

- **Development:** Candidate source revision, development state, exploratory results.
- **Test:** Reviewed effective configuration, test state, approved acceptance protocol.
- **Production:** Authorized effective configuration, production state, release evidence.
- **Cross-environment mapping:** Logical resource names linked to each environment's actual IDs.
- **Retention:** Protected state history and evidence references for each release.

This is an analyst design recommendation, not a documented multi-environment abstraction supplied by the CLI. Controlled systems documentation needs its own revision and change-control procedures under Part 11 where applicable. (www.ecfr.gov)

Serialize every writer that can affect the environment. The CLI README states that apply does not guard against concurrent execution. (github.com) GitHub concurrency limits jobs sharing the same group, so all participating writers need the same policy. (docs.github.com) A CI setting alone should not be assumed to cover laptops, another repository, or emergency tooling.

Drift, Destructive Flags, and Partial Failure

Treat drift as a decision requiring evidence

The applier refuses a plan containing blocked changes before sending operations. (github.com) A proposed drift response should begin with preservation: capture the blocked plan and obtain the current remote configuration through an authorized process. Determine whether the source, external change, target identity, or prior incomplete run explains the discrepancy. Proposed force handling should be an explicit exception, supported by the risk evaluation that ICH Q10 recommends for changes. (database.ich.org)

A practical decision sequence is:

- **Wrong target:** Stop and correct identity or state selection.
- **Approved external change:** Review whether repository declarations should incorporate it.
- **Unapproved divergence:** Preserve evidence and follow the organization's deviation procedure.
- **Missing remote resource:** Assess whether replacement changes downstream identity.
- **Uncertain earlier execution:** Reconcile remote observations with saved state.
- **Authorized recovery:** Produce a new plan and approve the resulting operations.

The sequence is proposed guidance. Its rationale is risk-based evaluation of the actual change and postimplementation confirmation, both described in ICH Q10. (database.ich.org) (database.ich.org) A drift check is useful technical evidence, but it does not itself classify quality impact.

Review flags by the effect they authorize

Table 2 connects operational flags to their specific review questions. These are proposed governance restrictions, not claims that the CLI supplies an approval system.

Option or action	Documented behavior	Proposed control
--dry-run	Plan-only mode exits 0 even when blocked. (platform.claude.com)	Classify plan content separately; do not use exit status as acceptance.
--yes with a directory	Noninteractive apply requires confirmation bypass; bare tracked-file reconciliation can skip new additions. (platform.claude.com)	Approve the exact source set and command before execution.
--force	Permits proceeding after external resource changes. (github.com)	Require preserved drift evidence and a specific exception decision.
--prune	Removes tracked resources outside the loaded source set. (github.com) Archives most kinds and deletes skills. (github.com) Vault pruning permanently discards credential secrets. (github.com)	Review the complete removal set and recovery prerequisites.
--upgrade	Resolves a newer external skill commit rather than retaining the prior pin. (github.com)	Treat the resolved dependency revision as a release change.
Source rename	Declares another resource and leaves the former resource until pruning. (platform.claude.com)	Assess identity, dependencies, duplicate operation, and retirement explicitly.

The important separation is between command authorization and quality acceptance. A reviewed source diff may still produce a plan containing a replacement, dependency upgrade, or removal requiring additional review. Changes affecting configuration should remain within the controlled procedure described by Annex 11. (health.ec.europa.eu)

A proposed policy should prohibit implicit force or prune on retry, require a full source inventory for removal operations, and retain the decision that authorizes each exceptional action. For vault changes, the evidence package should contain provisioning metadata and responsible roles without secret values. Access permissions should follow the minimum-necessary approach described by OWASP. (genai.owasp.org)

A failed apply needs reconciliation, not a success label

The implementation persists state after each individual success and stops at the first failure. (github.com) A proposed pipeline must retain both the changed lockfile and failure output, then hold the release pending reconciliation. The environment may contain resources from different stages of the intended change set.

Cancellation creates an additional boundary: implementation comments state that a write may or may not have reached the server. (github.com) Preserve the job result and query the actual resource before deciding what to retry. Do not infer remote nonexecution merely because a CI step was interrupted.

A proposed failure decision record should identify successful operations, uncertain operations, untouched dependencies, runtime sessions already started, and the chosen completion or restoration path. ICH Q10 recommends checking after implementation whether change objectives were achieved. (database.ich.org) That check should reference observed remote state and test results, rather than only the process exit code.

Evidence Packaging and Regression Design

Build a release record around observed outcomes

An **evidence package** should be readable by a system owner who did not participate in the deployment. It needs to connect intent, authorization, execution, and acceptance. SLSA describes provenance as verifiable information about where, when, and how an artifact was produced. (slsa.dev) That model is useful for release evidence, while pharmaceutical record requirements still require a separate applicability assessment. (www.fda.gov)

Table 3 lists a proposed package. Each retained item should have a responsible owner and an established retention location.

Evidence item	Content to retain	Question it answers
Intended use and impact	Workflow boundary, affected records, risk assessment. (www.fda.gov)	Why is this level of assurance appropriate?
Source and dependencies	Reviewed revision, effective files, resolved inputs. (slsa.dev)	What configuration was authorized?
Plan and approval	Detailed plan, blockers, approver decision, approved target. (www.ecfr.gov)	Who authorized which proposed change?
Identity and execution	Workflow identity, target workspace, command, timestamps, outcomes. (docs.github.com)	Which execution acted on which environment?
Before-and-after state	Lockfiles and authorized remote observations.	What changed successfully, and what remains uncertain?
Validation results	Test set, metrics, environment, deviations, acceptance decision. (airc.nist.gov)	Does observed behavior meet the intended use?
Provenance verification	Expected signer, builder, identity, and issuer checks. (slsa.dev) (docs.sigstore.dev)	Is evidence attributable to the expected production path?
Recovery and closure	Restoration exercise, residual differences, final review. (database.ich.org)	Can operation resume under an accepted configuration?

The package is intentionally broader than repository history. Part 11 also addresses systems-documentation revision controls; a deployment record should identify how those controlled documents relate to the executable release. (www.ecfr.gov) For record-producing workflows, document the risk assessment supporting the separate content-retention boundary. FDA recommends justified, documented assessment for validation decisions. (www.fda.gov)

Where artifact signing is adopted, verify the expected identity and issuer, not only the presence of a signature. (docs.sigstore.dev) Cosign supports verification of file signatures using a bundle containing a certificate and signature. (docs.sigstore.dev) This is a possible external evidence mechanism, not a claim that ant apply signs its plan.

Test behavior and permissions separately from configuration

A proposed regression protocol should cover instruction-following, retrieval from approved sources, citation accuracy, refusal to perform unauthorized actions, escalation, and handling of unavailable tools. Annex 11 calls for evidence of appropriate test methods and scenarios. (health.ec.europa.eu) Keep expected results tied to the approved workflow and assess the adequacy of the test environment. For applicable manufacturing computer systems, section 211.68(b) links input/output verification frequency and degree to complexity and reliability. (www.ecfr.gov)

Suggested test families are:

- **Task behavior:** Representative inputs, boundary cases, and expected reviewer outcomes.
- **Access:** Intended access succeeds; unauthorized operations do not proceed.
- **Dependencies:** Agent, skill, and environment combinations behave as approved.
- **Failure handling:** Tool unavailability and interrupted operations lead to accepted outcomes.
- **Records:** Required evidence can be exported, retained, and reconstructed.
- **Recovery:** Restoration produces an accepted operational configuration.
- **Ongoing checks:** Production monitoring detects changes in accepted behavior.

NIST recommends predeployment and operational testing, with documented methods and uncertainty measures. (airc.nist.gov) (airc.nist.gov) Its framework is voluntary; it should complement the applicable pharmaceutical quality requirements rather than be presented as a binding regulation. (www.nist.gov)

Release approval and runtime approval are different controls. Anthropic documents that custom-tool execution falls outside its runtime permission policies. (platform.claude.com) Proposed acceptance tests should therefore examine downstream authorization and human approval at the actual action boundary. OWASP recommends minimum necessary permissions and human approval for high-impact actions. (genai.owasp.org) (genai.owasp.org)

Design restoration outside ant apply

A proposed restoration design should preserve a known-good source revision, resolved dependencies, environment mapping, and separate runtime-data recovery procedures. Its final acceptance should follow postimplementation evaluation, as ICH Q10 recommends. (database.ich.org)

Restoration should be rehearsed as a newly authorized change. Assess whether republishing an earlier configuration creates a new version or identity, whether dependencies still exist, and whether memory or credentials require separate recovery. The proposed procedure should test configuration restoration and record restoration separately, retaining a justified assessment of record consequences. FDA's documented-risk-assessment recommendation provides the relevant validation basis. (www.fda.gov)

Operational stopping also requires a plan. Pausing a scheduled deployment does not terminate sessions already running. (platform.claude.com) A proposed recovery exercise should establish how future triggers, active sessions, downstream actions, and record capture are handled. Its success criterion should be accepted operating behavior and preserved evidence, consistent with postimplementation review. (database.ich.org)

Pilot Design and Operating Decision

A proposed pilot should use an isolated workflow with a defined intended use and a bounded set of authorized actions. The purpose is to establish whether the organization can control promotion and reconstruct outcomes. Risk-management effort should be proportionate to the consequence of the use, as ICH Q9 describes.

(www.ema.europa.eu)

A practical readiness checklist is:

- **Ownership:** Name the application owner, quality reviewer, and release operator.
- **Scope:** Identify records, business actions, and applicable procedures.
- **Data classification and eligibility:** Classify pilot inputs and records before admission. Managed Agents is currently ineligible for Zero Data Retention and HIPAA Business Associate Agreement (BAA) coverage. ([Anthropic Managed Agents overview](#)) Exclude workflows requiring either arrangement unless eligibility changes and is verified against current first-party documentation and applicable contract terms. These are conditional data-handling requirements; assess GxP validation separately for the intended use.
- **Retention:** Managed Agents session transcripts persist until deletion. ([Anthropic API and data retention](#)) Before admitting pilot data, review the vendor's server-side retention of conversation history, sandbox state, and outputs, and document acceptable retention, export, and deletion controls. ([Anthropic Managed Agents overview](#))
- **Version:** Freeze the qualified CLI and dependency resolution approach.
- **Identity:** Test intended workspace selection and rejection of unintended access.
- **Planning:** Demonstrate that a blocked dry run prevents promotion.
- **Approval:** Tie the approver's decision to the effective source revision.
- **Serialization:** Demonstrate control over every authorized deployment writer.
- **Failure:** Exercise partial execution and ambiguous cancellation.
- **Evidence:** Reconstruct a release from retained records alone.
- **Restoration:** Demonstrate accepted behavior after an authorized recovery change.

The checklist is proposed. FDA's risk-assessment recommendation supports selecting validation depth according to the relevant record and quality consequences. (www.fda.gov) It does not establish that completing a checklist validates the system.

The build-versus-manual decision should compare the organization's actual evidence burden. A manual process may suit a narrowly bounded experiment if approvals, identity mapping, and execution records remain reproducible. Repository-driven deployment becomes attractive when repeated changes and dependent resources make those mappings hard to maintain consistently. Either approach still needs controlled procedures for configuration changes. (health.ec.europa.eu)

Do not automate a weak approval boundary merely because configuration is reviewable. Downstream tools should receive only necessary permissions, and high-impact actions should have an assessed human approval boundary. (genai.owasp.org) (genai.owasp.org) The pilot should demonstrate these controls in operation, not only in a design document.

For laboratory uses, MHRA's page explicitly points to OECD guidance taking precedence in the Good Laboratory Practice (GLP) context. (www.gov.uk) A manufacturing pilot should not import laboratory-specific guidance as though it were a universal manufacturing requirement. Likewise, WHO's overview is not a binding regulatory framework. (www.who.int)

Data Analysis and Evidence

What the verified chronology establishes

The evidence establishes a command introduction, subsequent implementation changes, and a continuing beta platform designation. **September 3, 2026**, and CLI **1.30.0** are the launch baseline. (platform.claude.com) The opened changelog's latest entry was **1.40.0**, dated **October 9, 2026**, with a path-resolution change relevant to predefined agent lists. (github.com) Release qualification should therefore retain the actual tested binary version and source revision.

Regulatory-source chronology is separate. The opened Annex 11 revision states an operational date of **June 30, 2011**. (health.ec.europa.eu) The Commission's newer consultation page says its response period has ended and labels the materials as drafts; it does not establish final adoption. (health.ec.europa.eu) EMA identifies the current ICH Q9 version as effective from **July 26, 2023**. (www.ema.europa.eu) The distinction matters because a proposal should not be cited as an adopted requirement.

These dates do not provide a deployment-success rate, validation-effort estimate, or compliance score. Searches conducted for this report did not establish a directly relevant published pharmaceutical ant apply validation benchmark. The conclusion is limited to this research. Organizations should measure their own process and behavioral results; NIST recommends documented testing under conditions similar to deployment. (airc.nist.gov)

Calculate validation impact from reader inputs

Use a resource-to-test mapping rather than an invented industry average. The proposed calculation takes only reader-supplied values:

- **C:** Directly changed resources identified from the plan.
- **A:** Affected dependent resources identified by impact review.
- **T(r):** Applicable approved test cases for each resource *r*.
- **M:** Shared integration, record, permission, and recovery tests.
- **Required suite:** The union of T(*r*) for resources in C or A, plus M.

If every resource has the same applicable test count *k*, the initial mapping count is *k* multiplied by the size of the union of C and A, plus shared tests. Deduplicate shared cases before execution. This is a planning calculation, not a prediction of hours, cost, or validated status. The dependency set must include effective downstream changes, not only directly edited files.

The quality function should document why tests are included or excluded. ICH Q10 recommends risk-based evaluation of proposed changes; Annex 11 asks for evidence of appropriate scenarios. (database.ich.org) (health.ec.europa.eu) NIST additionally recommends documenting test sets, metrics, and tools. (airc.nist.gov)

Measure planned cases, executed cases, accepted results, unresolved deviations, and repeatability for the selected workflow. Record the denominator and acceptance rule rather than reporting a percentage without context. Where repeated model runs are evaluated, retain observed variability and uncertainty alongside the aggregate result, consistent with NIST's measurement approach. (airc.nist.gov) No universal acceptable threshold is proposed here.

The evidence model should also count unreviewed effective changes and incomplete resource outcomes. These are reader-defined operational measures. They become useful only when the source set, dependency boundary, and acceptance protocol remain identifiable. SLSA's input and provenance concepts provide a useful engineering vocabulary for that linkage. (slsa.dev) (slsa.dev)

Implications and Future Directions

The durable architectural issue is the boundary between declared infrastructure and evolving operational state. Repository review can govern configuration, but release acceptance must still assess the agent's intended behavior, records, and access. WHO's regulatory-considerations publication is a high-level overview, not an adopted regulatory framework. (www.who.int) (www.who.int) The same care should apply when interpreting vendor terminology about managed execution or enterprise controls.

The reviewed sources did not establish automatic adoption of arbitrary existing resources, built-in rollback, signed saved plans, or global deployment locking. Console exports with an included lockfile are a documented route to managing exported resources, rather than evidence of unrestricted adoption. (platform.claude.com) Every proposed deployment writer should enter the environment's coordination policy. Recheck control boundaries when upgrading and revise the acceptance protocol if behavior changes, keeping configuration changes within the defined procedure. (health.ec.europa.eu)

Future capability review should be tied to material effects: new resource kinds, dependency-resolution changes, identity behavior, state schema, permissions, and recovery operations. The proposed procedure should require an impact assessment before those changes enter a regulated environment. ICH Q9's proportionate-risk principle offers the relevant quality-management basis. (www.ema.europa.eu)

IntuitionLabs describes consulting, implementation, and custom development across Veeva platforms. (intuitionlabs.ai) In this context, an adjacent advisor's contribution is integrating agent delivery with existing record systems and quality procedures. The agent platform should fit the established ownership and approval model, with application and infrastructure responsibilities assessed separately as Annex 11 describes. (health.ec.europa.eu)

A useful next step is a pilot that proves evidence reconstruction and restoration before expanding workflow scope. Production monitoring should then test continuing fitness for intended use, rather than assume that a successful initial deployment settles future behavior. NIST explicitly recommends testing while systems operate. (airc.nist.gov)

Conclusion

ant apply can support a reviewable configuration-promotion process for Claude agents, but a pharmaceutical release decision needs more than repository files and a successful command. The proposed blueprint connects source review, environment identity, dependency impact, independent authorization, execution evidence, behavioral acceptance, and recovery. Its quality-management foundation is controlled configuration change and risk-based evaluation. (health.ec.europa.eu) (database.ich.org)

The most consequential implementation choices are explicit plan assessment, separate environment state, serialized writers, and preservation of outcomes after partial execution. A release should remain open when actual results differ from the authorized plan. Postimplementation evaluation is the point at which the organization determines whether its objectives were achieved. (database.ich.org)

The lockfile is useful state evidence within that process. It should not be presented as a validated audit trail or rollback system. Applicable electronic-record controls have their own requirements, and PIC/S distinguishes audit trails from change control. (www.ecfr.gov) (picscheme.org) External provenance, approval, retention, and recovery mechanisms must be assessed in the context of the intended use.

The pilot decision should be based on a demonstrable ability to reconstruct and restore accepted operation. If the team can connect an approved revision to its resource outcomes, applicable tests, and retained records, repository-driven promotion has a clear operational role. If those mappings remain ambiguous, the next investment should be in the missing control boundary. Documented test sets and deployment-representative assessment provide the appropriate evidence basis for that decision. (airc.nist.gov) (airc.nist.gov)

Source links

Links cited in this article, in order of first appearance. Full addresses are provided for readers using extracted text.

1. <https://platform.claude.com/docs/en/release-notes/overview>
2. <https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application>
3. https://health.ec.europa.eu/document/download/8d305550-dd22-4dad-8463-2ddb4a1345f1_en?filename=annex11_01-2011_en.pdf
4. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/core/lock.go>
5. <https://platform.claude.com/docs/en/cli-sdks-libraries/cli/apply>
6. <https://airc.nist.gov/airmf-resources/airmf/5-sec-core/>
7. https://database.ich.org/sites/default/files/Q10_Guideline.pdf
8. <https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11/subpart-B/section-11.10>
9. <https://picscheme.org/layout/document.php?id=714>
10. <https://slsa.dev/spec/v1.2/provenance>
11. <https://intuitionlabs.ai/articles/claude-managed-agents-gxp-approval-control-matrix>
12. <https://claude.com/resources/articles/claude-managed-agents>
13. <https://platform.claude.com/docs/en/managed-agents/overview>

14. <https://www.gov.uk/government/publications/guidance-on-gxp-data-integrity>
15. <https://intuitionlabs.ai/articles/risk-based-ai-validation-pharma-ich-q9>
16. https://www.ema.europa.eu/en/documents/scientific-guideline/international-conference-harmonisation-technical-requirements-registration-pharmaceuticals-human-use-ich-guideline-q9-r1-quality-risk-management-step-5-revision-2_en.pdf
17. <https://intuitionlabs.ai/>
18. <https://www.who.int/publications/i/item/9789240078871>
19. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/claude/skill.go>
20. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/claude/kinds.go>
21. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/claude/prose.go>
22. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/core/graph.go>
23. <https://github.com/anthropics/anthropic-cli>
24. <https://slsa.dev/spec/v1.2/build-provenance>
25. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/core/hash.go>
26. <https://nvlpubs.nist.gov/nistpubs/ai/NIST.AI.600-1.pdf>
27. <https://docs.github.com/en/actions/reference/workflows-and-actions/deployments-and-environments>
28. <https://platform.claude.com/docs/en/manage-claude/workload-identity-federation>
29. <https://docs.github.com/en/actions/reference/security/oidc>
30. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/claude/origin.go>
31. <https://www.ecfr.gov/current/title-21/chapter-I/subchapter-C/part-211/subpart-D/section-211.68>
32. <https://docs.github.com/en/actions/how-tos/deploy/configure-and-manage-deployments/control-deployments>
33. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/core/apply.go>
34. <https://github.com/anthropics/anthropic-cli/blob/main/internal/declarative/core/plan.go>
35. <https://genai.owasp.org/llmrisk/llm062025-excessive-agency/>
36. <https://intuitionlabs.ai/articles/pharma-ai-validation-evidence-fda-ema>
37. <https://docs.sigstore.dev/quickstart/verification-cheat-sheet/>
38. <https://docs.sigstore.dev/cosign/verifying/verify/>
39. <https://www.nist.gov/publications/artificial-intelligence-risk-management-framework-generative-artificial-intelligence>
40. <https://platform.claude.com/docs/en/managed-agents/permission-policies>
41. <https://platform.claude.com/docs/en/managed-agents/scheduled-deployments>
42. <https://platform.claude.com/docs/en/manage-claude/api-and-data-retention>
43. <https://github.com/anthropics/anthropic-cli/blob/main/CHANGELOG.md>
44. https://health.ec.europa.eu/consultations/stakeholders-consultation-eudralex-volume-4-good-manufacturing-practice-guidelines-chapter-4-annex_en
45. <https://www.ema.europa.eu/en/scientific-guidelines/ich-q9-quality-risk-management>

THE PUBLISHER

About IntuitionLabs

IntuitionLabs is an AI consulting, custom software development and data engineering firm serving pharmaceutical, biotechnology, medical-device and other life-science organizations. We work with clinical, regulatory, medical-affairs, commercial, quality and IT teams to connect technology decisions with the work people need to accomplish.

AI consulting and adoption

Our AI enablement services cover readiness assessments, use-case selection, governance and policies, team workshops, adoption measurement and ongoing advisory support. We help organizations structure the information layer behind AI: source material, context, permissions and maintained knowledge that make generated answers useful and reviewable. Private LLM inference and hosted AI options support teams evaluating how to operate AI with appropriate control over their data and infrastructure.

Software, data and life-science workflows

IntuitionLabs develops custom software for pharma and biotech, integrates enterprise systems, and builds data engineering and business intelligence solutions. Areas of focus include AI agents, regulatory research, medical writing, medical affairs, CMC information, competitive intelligence and clinical-document workflows. Our eTMF intelligence work includes cross-system reconciliation and inspection-readiness support.

Enterprise platforms and regulated delivery

We provide Veeva services, application support, managed services, integrations and custom applications, alongside enterprise content work involving platforms such as Egnyte. For regulated workflows, our services include GxP enablement, computer-system validation and software development addressing 21 CFR Part 11 requirements. The applicable controls, validation responsibilities and acceptance criteria are defined for each engagement.

Work with IntuitionLabs

Explore [AI enablement](#), [pharma and biotech software development](#), [data engineering and BI](#), and [Veeva services](#). [Contact IntuitionLabs](#) to discuss your workflow, information sources and implementation needs.

IntuitionLabs publishes educational research to help life-science teams make informed technology decisions. Coverage of a product or organization does not imply a client relationship, endorsement or partnership.

Website: <https://intuitionlabs.ai>

This article is educational. Verify consequential medical, legal, financial, regulatory and technical decisions with appropriate professionals and the cited primary sources. Company services are described separately from the article's research findings.