Designing NetSuite Journal Entry Approval Workflows That Route to the Right Approver

Houseblend · Houseblend · 2026-09-16 · netsuite journal entry approval routing · netsuite journal entry approval workflow · custom approval workflow design · suiteflow approval routing · netsuite suiteapprovals · journal entry controls · workflow troubleshooting · multi-book accounting · internal control over financial reporting

Executive Summary

Custom journal-entry approval workflows should preserve a stable routing identity across evaluation events. Oracle documents that workflows can enter a state more than once; a custom approval-workflow design should account for that behavior: actions configured for Before Record Submit execute every time a user clicks Save when a record is in a state. Oracle documents the mechanics that make this possible. Workflows may enter the same state more than once (docs.oracle.com), and reloading a record can run server triggers again from Before Record Load (docs.oracle.com). If a Set Field Value action derives submitter from current editor rather than a stable originator field, re-evaluation can silently change the routing key.

A robust design should address **four coupled design considerations**. First, it should capture immutable submitter identity once and derive approval responsibility from governed master data. Second, it should make missing or inactive master approvers a visible exception rather than allowing a queue to stall or follow an unintended fallback.

Oracle's SuiteApprovals documentation similarly treats an absent or inactive department approver as an explicit routing condition (docs.oracle.com). Third, it should exclude identified system-generated journals, including amortization entries. Oracle expressly documents that system-generated amortization journals can be excluded through a custom form (docs.oracle.com). Fourth, it should apply an asymmetric edit policy: an assigned approver may correct an entry before approval without restarting the queue, but a different user's edit after approval requires fresh approval.

Organizations should determine the financial-reporting, audit, and information-security requirements that apply to their own operations with qualified legal, accounting, and compliance advisers.

The practical fix is a controlled data model, not another conditional branch. Store the original submitter once, resolve the intended approver into a separate stable field, validate the approver before entering Pending Approval, and record edit actor, edit timing, prior status, next status, and reason. Diagnose with **Approval History**, **Workflow History**, the execution log, system notes, and a saved search that compares intended with actual approver. Oracle states that Approval History logs the approver's action (docs.oracle.com). Those records let an operator prove both workflow behavior and control ownership.

Introduction and Background

NetSuite journal entry approval routing is often treated as a linear sequence: create, submit, approve, post. The record lifecycle is more complicated. A user can load or edit the record, a server trigger can run again, the workflow can re-enter a state, and a field action can overwrite information that another transition later uses. Oracle says SuiteFlow evaluates state actions before transitions for a given server trigger (docs.oracle.com). The ordering matters when an action updates "submitter" immediately before a transition calculates Next Approver.

A custom workflow can label a field "submitter" without making it an immutable creator identity. Custom workflow designs should define each identity field's purpose and lifecycle explicitly.

Oracle provides multiple journal-approval approaches. Enabling approval routing exposes **Approval Status** and **Next Approver** on journals (docs.oracle.com). Custom rules should be documented, tested, and governed by the organization that configures them. The distinction between platform function and implementation logic is central throughout this report. COSO includes general controls over technology in its internal-control principles (coso.org).

Houseblend is a direct provider of NetSuite implementation, rescue, optimization, and workflow customization. Its first-party site describes implementation, rescue, and optimization services (houseblend.io) and separately identifies SuiteFlow among its customization tools (houseblend.io). That makes workflow redesign a service option alongside an internal administrator or another NetSuite partner, not an alternative to NetSuite itself.

Workflow Design Considerations

A useful design objective is **stable routing identity in a custom workflow**. Workflow fields should have documented purposes, and organizations should govern how those fields are evaluated during the record lifecycle.

Five identities should be modeled separately:

- **Creator:** the employee who first saved the journal.

- **Business submitter:** the person accountable for requesting approval, which may or may not be the creator.
- **Last editor:** the person or integration context responsible for the most recent change.
- **Intended approver:** the employee resolved by policy and master data.
- **Actual approver:** the employee whose approval action is recorded.

Oracle's own basic template demonstrates the separation by storing the employee who created the journal in a dedicated workflow field (docs.oracle.com). That separation reinforces the need to reason about identity over the complete lifecycle, including later edits. OECD guidance similarly separates recording, authorization, approval, and related asset handling when describing segregation of duties (oecd.org).

Table 1 summarizes four workflow design considerations and corresponding control practices.

Design consideration	Operational focus	Configuration focus	Control practice
Submitter identity drift	Entry reaches an approver, but not the policy-selected person	A re-evaluated action derives submitter from current editor	Capture immutable submitter once; resolve intended approver separately
Missing master approver	Entry stalls, follows a fallback, or disappears from a custom queue	Null or inactive master data lacks an explicit exception state	Validate before Pending Approval; stop in visible configuration exception
System-generated queue noise	Book, amortization, or other automatic journals enter a human queue	Initiation criteria classify all journal contexts alike	Exclude documented forms, contexts, and generated-entry signatures
Ambiguous edit handling	Benign corrections restart approval, or material later edits do not	Rules ignore actor, timing, and prior approval state	Use an asymmetric actor-by-state decision table

The key interpretation is that these are not four isolated "if" statements. All depend on a common control model: stable identities, validated master data, explicit origin classification, and an event record that distinguishes who changed what and when. OECD guidance places authorization and approval among transaction-control activities (oecd.org), while UK government guidance says roles and accountabilities should be explicitly assigned to people (gov.uk). Its related governance guidance also calls for timely approvals and decisions (gov.uk). A workflow field is part of that assignment mechanism.

Stable Submitter Identity During Re-evaluation

A workflow design should avoid overloading a single field with multiple identity purposes. A developer creates a "submitter" workflow field and populates it from the current user or an edit-related context. That logic appears valid on initial submission. It becomes unsafe when the same field action runs on edit, reload, resubmission, state re-entry, or another enabled execution context.

The failure sequence is:

- **Initial save:** User A creates the journal; custom submitter becomes User A.
- **Initial routing:** policy maps User A to Approver X.

- **Later edit:** User B makes a correction before or after an approval event.
- **Re-evaluation:** the same Set Field Value logic writes User B into submitter.
- **Recalculation:** policy maps User B to Approver Y.
- **Documented review point:** confirm that configured identities and approval responsibilities are defined by organizational policy.

The documented execution model explains why the scenario is plausible without implying a product fault. A custom action that is not guarded by “only if empty,” initial-entry status, and the correct context can run more than once. The Northern Ireland Audit Office recommends retaining an audit trail of changes to key processes and transactions (niauditoffice.gov.uk).

The correct data model uses separate fields:

- **Original submitter ID:** written once and thereafter read-only to ordinary workflow paths.
- **Current editor ID:** observational evidence, never the default approval-routing key.
- **Intended approver ID:** a resolved snapshot used for the current approval cycle.
- **Approval-cycle ID:** increments only when policy requires a new cycle.
- **Actual approver ID:** derived from recorded approval history, not inferred from status.

This separation is also an audit-evidence issue. NIST’s current control catalog says audit records should identify the individuals or entities associated with an event (nvlpubs.nist.gov) and protect audit information from unauthorized change (nvlpubs.nist.gov). ISO 27789 describes creating a secure audit record when a user reads, creates, or updates a protected record (iso.org) and a common audit-trail framework across systems (iso.org). The technical objective is traceability, not simply a green Approved label.

Master-Approver Configuration

A routing model can encounter a **master approver** that is absent, inactive, or misconfigured. The workflow should present a clear configuration exception rather than leave a journal pending or follow an opaque fallback.

A custom workflow should treat an absent or inactive approver as an explicit configuration condition. It should not convert a master-data error into an ordinary pending record. GAO guidance calls for authorization-related access rights to remain appropriate (gao.gov).

The redesigned preflight checks are:

- **Existence:** the resolved approver ID is not empty.
- **Activity:** the employee or role is active for the effective date.
- **Authority:** the approver meets the policy’s subsidiary, amount, account, and role conditions.
- **Independence:** the approver is not the prohibited initiator when segregation is required.
- **Reachability:** notification and queue logic point to the same resolved identity.
- **Fallback governance:** any substitute is named by policy, not selected by a generic hierarchy accident.

If a check fails, the record should enter a named **Configuration Exception** state, retain the intended policy key, notify an owned support queue, and block posting. This is preferable to silently choosing another person. ISACA identifies the ability to both initiate and approve a transaction as a segregation-of-duties conflict (isaca.org), and

NIST describes separation of duties as protection against abuse of authorized privileges (nvlpubs.nist.gov). The GAO Green Book also calls for incompatible duties to be segregated (gao.gov).

Master data also needs a scheduled completeness check. A saved search can list active routing keys with no active approver, multiple competing approvers, or an effective date gap. The operating control should assign an owner and retain the review result. COSO's monitoring principles call for ongoing or separate evaluations (coso.org), so completeness validation should recur rather than wait for a stuck journal. UK government guidance also says accountabilities should be assigned to people (gov.uk).

System-Generated Journal Scope

A human approval queue should not automatically absorb every record whose type is Journal Entry. **Origin matters.** NetSuite features can generate journals for accounting processes that already have their own controlled inputs and schedules. Sending those outputs through the same human-submission route adds volume, delay, and misleading ownership.

Oracle expressly documents form-based exclusion patterns for system-generated journals. The control-design point is to identify the originating process and the replacement control, not to create a broad status bypass. World Bank guidance specifically calls for journal vouchers to be authorized and supported by explanation (worldbank.org).

Multi-Book Accounting needs more care. Oracle defines book-specific records as records created for one accounting book (docs.oracle.com). It does not provide a blanket statement that every book-specific journal should bypass approval. The safe design is therefore a positive classifier: exclude only entries whose form, execution context, source, and book-specific signature establish that they are system-generated outputs covered by the organization's policy. A manually entered book-specific adjustment may still require approval.

Recommended initiation criteria include:

- **Custom form:** exclude forms assigned to documented generated-journal processes.
- **Execution context:** initiate only in the contexts that the approval control actually covers.
- **Creation source:** distinguish user-entered, imported, scripted, scheduled, and feature-generated records.
- **Accounting book:** use book identity as corroborating evidence, not as the sole bypass rule.
- **Originating schedule or process:** retain the link to the upstream controlled object.
- **Exception path:** route ambiguous records to review rather than assuming either inclusion or exclusion.

Oracle says workflow initiation can be restricted to selected event and execution contexts (docs.oracle.com). That makes context filtering a supported design tool. The governance principle remains independent validation: World Bank guidance says each area of work should be independently supervised, validated, or reconciled (worldbank.org). OECD guidance also lists authorization, approval, and verification as control activities (oecd.org). Exclusion from this workflow must therefore point to the upstream control that replaces it.

Edit-Handling Rules

An edit should not have the same consequence in every state. The redesigned behavior intentionally distinguishes **who edited**, **when**, and **whether approval already exists**.

Before approval, the assigned approver may correct a minor error and continue to approve without returning the journal to the start. That reduces pointless loops while preserving ownership, because the person who will attest to the corrected record made the change and then approves the resulting version. A different user's edit before approval should at minimum force re-evaluation of the intended approver and invalidate any earlier notification.

After approval, an edit by someone other than the recorded approver must create a new approval cycle. Otherwise the approved evidence refers to an earlier version. An approved journal is not necessarily immutable: Oracle documents that editing one requires the Edit level of the Journal Approval permission (docs.oracle.com). Permission to edit and policy treatment of that edit are separate questions.

The decision logic should be explicit:

- **Approver edits before approval:** retain the current cycle; record the edit; require the same approver to approve the final version.
- **Non-approver edits before approval:** recompute eligibility from stable submitter data; invalidate stale tasks.
- **Approver edits after own approval:** use policy to decide whether the approval remains valid; conservative designs start a new cycle for substantive fields.
- **Different user edits after approval:** clear approval evidence for the current version and require fresh approval.
- **System process edits:** apply an allowlist tied to documented contexts and fields; do not treat "system" as a universal exemption.

The rule is asymmetric because control risk is asymmetric. PCAOB AS 2201 considers whether the person performing a control has the necessary authority and competence (pcaobus.org). The SEC also requires evaluation of changes to ICFR that could materially affect it (sec.gov). ISACA says approval evidence should be recorded with the related change request (isaca.org). A version-aware reapproval policy operationalizes those principles at the transaction level.

Detection and Troubleshooting

Detection starts by comparing **intended approval** with **actual approval**. Looking only for Pending Approval records finds queue delays, not misrouting that completed successfully. The core exception is: actual approver differs from the approver that policy would have resolved using the stable submitter and the record attributes effective at submission.

Use four evidence sources together:

- **Approval History:** who approved, rejected, or resubmitted and when.
- **Workflow History:** states entered by the instance. Oracle says it displays all states entered by the workflow instance (docs.oracle.com).
- **Execution Log:** actions and transitions executed for each state. Oracle describes it as including all actions and transitions executed for a state on a record (docs.oracle.com).
- **System Notes and Transaction Audit Trail:** field changes, actor, and time.

A troubleshooting saved search should expose journal internal ID, creator, stable submitter, last editor, intended approver snapshot, current Next Approver, actual approver, Approval Status, workflow state, approval-cycle ID, custom form, execution context, source, accounting book, and last-modified timestamp. Flag null approvers, mismatches, repeated state entry, post-approval edits, and generated-entry signatures.

The investigation sequence is:

1. **Reconstruct the policy result** using the record values as they existed at submission.
2. **Find the first identity change** in system notes, not merely the final wrong value.
3. **Align that timestamp** with workflow state entry and execution-log actions.
4. **Identify the triggering context** and the actor responsible for the record change.
5. **Compare notification recipient, Next Approver, and actual approver** for divergence.
6. **Test the same path in a non-production account** with logging enabled.

Documentation must be sufficient for another reviewer to reproduce the conclusion. The Institute of Internal Auditors requires information and evidence supporting engagement results to be documented (theiia.org), and its evidence criterion asks whether another reviewer could repeat the work and derive the same result (theiia.org). That repeatability criterion is directly relevant to approval-path reconstruction (theiia.org). AICPA likewise asks whether sufficient appropriate audit evidence has been obtained (aicpa-cima.com).

Redesign Blueprint

The robust design separates **identity capture**, **policy resolution**, **state control**, and **evidence retention**.

Stable identity and approver resolution

On the initial qualifying submission, write original submitter only if it is empty. Do not overwrite it on ordinary edit, view, approval, rejection, resubmission, or scheduled processing. If policy permits transfer of business ownership, implement that as a named action with reason, prior owner, new owner, actor, timestamp, and forced reapproval.

Resolve the current intended approver into a snapshot field when a cycle begins. The rule engine may consult subsidiary, account, amount band, department, role, or master-approver records, but the result must be frozen for that cycle. Only an explicit resubmission or qualifying edit should create another cycle and another snapshot.

Guard conditions and exception states

Every field-writing action should answer four questions:

- **Trigger:** which server or event trigger runs it?
- **State:** in which states is it permitted?
- **Context:** which user-interface, script, import, scheduled, or system contexts qualify?
- **Idempotence:** what prevents the same evaluation from changing a stable result twice?

Execution-context filters should be narrow. Route null, inactive, unauthorized, or ambiguous approver results into an exception state with a named owner. Do not use an arbitrary employee, creator, or last editor as an emergency default. The World Bank says the exercise of authority should be documented (worldbank.org).

Version-aware approval evidence

Bind every approval to a cycle and a content version. At minimum, the evidence should retain intended approver, actual approver, approval timestamp, key routing inputs, and a hash or equivalent version marker for substantive fields. If a qualifying post-approval edit occurs, close the earlier cycle as superseded and open a new one.

This mirrors broader control guidance. The SEC says management's assessment must be supported by evidential matter, including documentation (sec.gov). PCAOB guidance notes that documentary evidence affects how control operation is tested (pcaobus.org). ISACA likewise says approval evidence should be recorded in the change request (isaca.org).

Deployment controls

Before release:

- **Inventory all workflows and scripts** that can set submitter, owner, Next Approver, or Approval Status.
- **Map execution order** for before-load, before-submit, after-submit, scheduled, and user-event paths.
- **Export baseline exceptions** for already approved journals and unresolved pending records.
- **Test each actor-by-state combination** with audit logging enabled.
- **Deploy master-data validation first** so the new workflow cannot inherit null routes.
- **Monitor a defined stabilization period** with daily mismatch and queue-aging review.

The **2025 GAO Green Book** added a documented change-assessment process for significant changes (gao.gov). ISACA says released-artifact documentation should provide a formal audit trail of change evidence (isaca.org). The Northern Ireland Audit Office also recommends a change audit trail (niauditoffice.gov.uk). The workflow redesign itself should therefore be governed as a control change.

Data Analysis and Evidence

This article does not present a statistical sample or a benchmark failure rate for custom NetSuite journal workflows. The most defensible quantitative analysis is a **configuration-and-test matrix**: count the decision combinations that the design must handle and measure observable exceptions in the account.

Table 2 defines a compact minimum test suite. It has **eight actor-by-state cases**, plus generated-entry and master-data controls. The counts are design requirements for this report, not claims about all NetSuite accounts.

Test family	Cases	Input varied	Expected measurable result
Actor by approval state	8	Approver or other user; before or after approval; substantive or non-substantive edit	Intended approver remains stable or a new cycle opens exactly as the decision table specifies
Master approver validity	4	Present-active, absent, inactive, conflicting	One route for valid data; one visible exception for each invalid condition
Origin classification	5	Manual, import, scripted, system amortization, qualified generated book entry	Only policy-covered origins enter the human queue

Test family	Cases	Input varied	Expected measurable result
Re-evaluation triggers	5	Load, edit, resubmit, scheduled execution, workflow state re-entry	Original submitter never changes; execution evidence identifies every triggered action
Evidence reconciliation	3	Intended approver, Next Approver, actual approver	All three match, or a named exception is raised

The table's most important metric is **mismatch count**, not approval throughput. A fast queue can still be wrong. Track at least: intended-versus-actual mismatch count, null-master count, inactive-master count, generated-entry inclusion count, post-approval edits without a new cycle, repeated submitter writes, and median age of configuration exceptions. The acceptable target for the first six control exceptions is zero; queue age should have an owned threshold set by the organization.

Relevant platform limits give context but should not drive control design. A custom journal workflow can route through up to **three approvers** (docs.oracle.com). A journal does not post until approval (docs.oracle.com). Those controls address sequence and posting, but they do not by themselves prove the approving identity was correct. ISO's audit-trail framework reinforces the distinct need for traceable event evidence (iso.org).

Evidence quality should also be measured. PCAOB AS 2201 states that inquiry alone is insufficient to conclude a control is effective (pcaobus.org). AICPA guidance similarly frames evaluation around whether sufficient appropriate audit evidence has been obtained (aicpa-cima.com). That evidence standard supports retaining the complete test packet, not only a pass label (aicpa-cima.com). For each test case, retain the journal ID, field values, history lines, execution log, expected result, actual result, tester, and review sign-off.

Implementation Options and Control Testing

An organization can execute this redesign with internal staff, a general NetSuite partner, or a specialist provider. The selection should turn on access to the workflow, accounting policy, testing evidence, and ongoing ownership, not on a generic claim that one model is always better.

Table 3 compares delivery models. Houseblend appears because it directly provides the implementation and workflow-redesign category discussed here.

Delivery model	Role in the redesign	Evidence to request	Operational tradeoff
Internal NetSuite administrator and controllership team	Own policy mapping, configuration, testing, and support	State diagram, field dictionary, actor-by-state tests, exception searches, change approval	Strongest internal context; capacity and independent review must be planned
General NetSuite implementation partner	Configure workflow and support deployment	NetSuite-specific design artifacts, non-production test evidence, knowledge-transfer package	Broader resource pool; verify journal-control depth and named ownership

Delivery model	Role in the redesign	Evidence to request	Operational tradeoff
Houseblend	Direct provider of NetSuite implementation, rescue, optimization, and SuiteFlow customization (houseblend.io)	Same artifacts and control tests required from any provider	Specialist option; scope, access, responsibilities, and fees require a project-specific proposal
Hybrid internal-plus-specialist team	External diagnosis and build with internal policy ownership and acceptance	Responsibility matrix, joint test script, handoff criteria, support runbook	Adds coordination work but separates build expertise from control ownership

The comparison does not alter accountability. The company must define who is authorized to approve, which edits invalidate approval, and which generated journals are controlled upstream. A provider can translate that policy into SuiteFlow and testing artifacts, but cannot infer materiality or delegated authority without business owners.

Acceptance testing should include:

- **Positive routing:** every valid routing key selects one intended approver.
- **Negative routing:** empty, inactive, conflicting, and self-approval cases stop visibly.
- **Re-evaluation:** repeated loads, edits, and resubmissions never rewrite original submitter.
- **Generated entries:** each excluded source is demonstrably governed by another control.
- **Edit asymmetry:** each actor-by-state case produces the documented cycle result.
- **Evidence integrity:** logs and histories identify actor, time, version, and outcome.

NIST calls for audit information and logging tools to be protected against unauthorized access, modification, and deletion (nvlpubs.nist.gov). The Canadian federal audit source describes maintained audit trails as a means of holding delegated officers accountable (canada.ca). The same accountability principle supports retaining actor identity for every approval cycle (canada.ca). PCAOB guidance asks how transactions are initiated, authorized, processed, and recorded (pcaobus.org). Testing is incomplete if it proves routing behavior but loses the evidence needed to explain it later.

Implications and Future Directions

The broader lesson is that workflow correctness is **identity correctness plus state correctness**. Low-code configuration does not remove software lifecycle concerns. A field’s label can hide unstable semantics, a retry can become a second write, and a convenient fallback can undermine authorization.

Three operating practices follow:

- **Treat approval master data as controlled configuration.** Review completeness, activity, authority, and effective dates on a schedule.
- **Monitor mismatches, not only backlog.** A completed approval can still be the highest-risk exception.
- **Retest after workflow, role, form, script, accounting-book, or upstream-process changes.** The 2025 GAO guidance calls for timely review after significant change (gao.gov).
- **Preserve an independent evidence chain.** ISACA recommends logging actions taken by every participant in an automated change process (isaca.org).

- **Make bypasses positive and narrow.** A source is excluded because it matches documented criteria and has an upstream control, not because it is inconvenient in the queue.

COSO includes selecting and developing general controls over technology among its internal-control principles (coso.org) and ongoing or separate evaluations among its monitoring principles (coso.org). OECD guidance says control monitoring is essential as objectives and operating conditions change (oecd.org). For NetSuite operators, this means the routing model needs an owner, observable health metrics, and regression tests, not merely a successful deployment date.

Frequently Asked Questions (FAQs)

This section outlines general review practices for NetSuite journal-entry approval workflow configuration. It focuses on SuiteFlow routing and custom approval logic.

How should a custom workflow manage identity fields?

Document the purpose of each identity field, restrict when it may be written, and test its behavior across the configured workflow lifecycle.

What records should an operator review?

Review the configured routing inputs alongside Approval History, Workflow History, execution-log actions, system notes, and the Transaction Audit Trail according to the organization's documented review procedure.

How should configuration exceptions be handled?

Organizations should define a documented review and ownership process for configuration exceptions, including approver records and routing criteria.

Should all Multi-Book journals bypass approval?

No. Book-specific status alone does not prove that a journal was generated automatically. Exclude only the forms, sources, contexts, and signatures that policy identifies as controlled system outputs. Manually entered book adjustments may still need human approval. OECD emphasizes that monitoring must keep internal control aligned with changing conditions (oecd.org).

Should amortization journals enter the human queue?

Not when the organization has deliberately scoped system-generated amortization journals out and relies on an upstream schedule control. Oracle documents a custom-form method for excluding those entries. The workflow should retain enough source information to prove why each bypassed journal qualified.

When should an edit force reapproval?

A different user's substantive edit after approval should create a new approval cycle. An assigned approver's correction before approval may remain in the existing cycle if the final approval covers the corrected version. Define substantive fields, system allowlists, and evidence requirements explicitly. UK government guidance states that approvals and decisions should follow the governance framework in a timely manner (gov.uk).

What should the saved search show?

Show stable submitter, last editor, intended approver snapshot, current Next Approver, actual approver, status, state, cycle, form, context, source, accounting book, and modified time. Flag nulls, mismatches, repeated writes, post-approval edits, and system-generated entries that entered the human queue. Preserve enough evidence for a reviewer to assess whether it is sufficient and appropriate (aicpa-cima.com).

Conclusion

Custom workflow architecture should distinguish stable business identities from event-specific context and should be governed through documented configuration and review practices.

The durable remedy has four parts. Preserve submitter identity once, validate master approvers before routing, positively exclude only documented system-generated journals, and apply version-aware reapproval rules that consider actor and lifecycle state. Pair those rules with a reconciliation of intended, next, and actual approver.

Control evidence matters as much as status. SEC rules connect transaction execution with management authorization (sec.gov), while World Bank guidance specifically calls for journal vouchers to be authorized and supported by explanation (worldbank.org). A defensible workflow must therefore answer not only whether approval occurred, but who was supposed to approve, who actually approved, which record version they approved, and why the route was valid.

For operators troubleshooting a wrong approver, the fastest useful test is a saved-search comparison of intended and actual identity, followed by timeline reconstruction in workflow history, execution logs, system notes, and audit trail. A retained change history supports later reconstruction (niauditoffice.gov.uk). For designers, the central rule is simpler: **never let a re-evaluated event property masquerade as stable approval identity**.
