

JIRA PROCESS & GOVERNANCE GUIDE

Portfolio Excerpt

Created by Jesse Molina | Senior Business Analyst | Jira Administrator

Original Context	Internal process guide created at Luckie & Company to standardize Jira usage and improve delivery discipline.
Purpose	Define a repeatable Jira operating model for project setup, ticket structure, QA/ME review, feedback handling, and UAT handoff.
Portfolio Treatment	Condensed from a 24-page internal guide. Client-specific URLs, board links, environment addresses, and internal identifiers have been removed or generalized.
Skills Demonstrated	Jira administration, workflow design, process governance, BA/QA collaboration, ticket taxonomy, release readiness, SOP documentation, change management.

1. Process Overview

The redesigned Jira process established a controlled path from build-kit intake through development, QA, editorial review, client feedback, UAT, and launch. The goal was to reduce inconsistent ticketing, unclear ownership, duplicated comments, and lost feedback while creating reliable traceability across teams.

- Build kit review by PM, BA, development, QA, and editorial teams
- Create Jira project structure, epic, version, and parent tickets
- Development begins using standardized ticket descriptions and references
- QA and editorial reviews occur in parallel after development completion
- Issues are captured as dedicated subtasks rather than buried in comments
- External feedback is normalized into proofs and linked Jira work items
- Completed changes are revalidated by QA and then moved to UAT
- Final smoke testing and release validation precede launch

Governance Principles

- One source of truth for actionable work
- Consistent naming conventions and hierarchy
- Clear ownership at every workflow stage
- Evidence attached directly to the relevant ticket
- Separate parent work from review work and issue-level subtasks
- Explicit handoffs between Dev, QA, editorial review, PM/BA, and UAT

2. Jira Board Setup Standards

Before ticket creation begins, the BA or PM establishes the project structure and ensures the work is associated to the correct epic and version. This prevents fragmented work and makes reporting more reliable.

Epic Setup

- Create the appropriate Jira project and epic
- Populate the epic name, summary, assignee, description, due dates, and project reference
- Move the epic out of backlog when ready for active use
- Associate all relevant parent tickets to the epic

Parent Ticket Standard

Parent tickets represent the highest level of execution work directly used by developers, QA specialists, and editorial reviewers. A parent ticket is typically created for each web page or major functional section receiving work.

Field	Standard
Naming	[Site Abbrev.] Parent [Section or Page] [Brief Work Description]
Description	Date + site/page context + concise work description
References	Link to approved proof/build kit or other delivery artifact
Dev Direction	Specific implementation direction and relevant source reference
QA Direction	Verification instructions tied to the requested change
Environment	Include only when needed for execution or validation

Portfolio note: live URLs, internal board IDs, and pre-production environment addresses from the original guide have been intentionally removed.

3. Managing the Jira Board

The revised process introduced stronger signals for blocked work and clearer views for individual ownership and team-level monitoring.

Flagging Rules

- Flag a ticket when work cannot continue without BA or PM input
- Flag when a cross-functional answer is required before execution can proceed
- Flag when a missing asset, approval, or dependency blocks work
- Remove the flag after the blocking condition is resolved

Useful Views

- Personal work view to track all assigned tickets
- Main board view for parent-level project status
- Flagged-ticket view for rapid triage of blockers and unanswered questions

Why This Matters

The intent was not simply to make Jira cleaner. The operating model was designed to make unanswered questions visible, keep work from stalling silently, and give PMs and BAs a practical way to prioritize intervention.

4. QA and Review Workflow

The process separated development work from review work while preserving linkage between them. This allowed QA and editorial teams to track their own workload without repeatedly passing a single ticket back and forth.

- 1 Create parent ticket plus QA review and editorial review tickets
- 2 Developer moves the parent ticket to In Progress
- 3 Developer moves completed work to Dev Complete
- 4 QA and editorial review tickets move from backlog to their active lanes
- 5 Each identified issue becomes a dedicated subtask linked to the relevant review ticket
- 6 After fixes are validated, subtasks close and review tickets return to the PM/BA for completion

QA Subtask Standard

- One identified issue per subtask
- Name includes date, site/page reference, and issue number
- Include device or screen size where the problem was found
- Attach screenshots or other visual evidence
- Provide specific reproduction or correction direction
- Return the completed subtask to QA for confirmation before closing

5. External Feedback and UAT Handoff

External partner feedback was converted into structured, actionable work rather than left as disconnected comments. PMs and BAs reviewed feedback together, created a shared proof, and translated each actionable item into traceable Jira work.

- Review feedback as a lead team to establish shared understanding
- Determine whether each comment is actionable, requires clarification, or should be rejected with rationale
- Create page-level parent tickets for the feedback round
- Create one subtask per actionable comment
- Link the supporting proof or design reference to the ticket
- Send completed work through QA validation before moving it to UAT
- Move the parent item to UAT only after all linked subtasks have been validated

What This Sample Demonstrates

- Operational process redesign, not just tool configuration
- Jira governance and administration in a cross-functional delivery environment
- Structured handoffs across Development, QA, editorial review, PM/BA, and UAT
- Translation of ambiguous feedback into actionable, traceable work
- Documentation designed to improve adoption and team consistency

This portfolio excerpt is intentionally condensed and sanitized from the original internal guide. It preserves the process design and documentation approach while removing sensitive internal links, environment details, board identifiers, and unnecessary client-specific operational information.