Sample Jira Workflow

This workflow assumes the use of a Scrum methodology and covers typical phases in software development: Requirement Gathering, Design, Development, Testing, and Release.

Workflow Overview

Statuses:

- Backlog: Items that are yet to be worked on.
- To Do: Approved items ready for work.
- In Progress: Items currently being worked on.
- Code Review: Development is complete; awaiting review.
- Testing: Undergoing QA testing.
- Blocked: Blocked items awaiting resolution.
- Done: Work completed and verified.

Transitions:

- Backlog → To Do: When an item is approved.
- To Do → In Progress: When a developer picks up the task.
- In Progress → Code Review: After completing development.
- Code Review → Testing: After code is approved.
- Testing → Done: Once QA confirms success.
- Any → Blocked: If issues arise.

Below is a detailed breakdown of issues across different stages of the workflow.

1. Requirement Gathering
Objective: Document user stories, bugs, and tasks.
Issues:

- Write user story for login functionality.
- Create a bug report for login errors.
- Define task for database schema design.
- Document API integration requirements.
- Create a user story for password recovery feature.

2. Design
Objective: Create wireframes, database schema, and API designs.
Issues:

- Design wireframe for login page.
- Design database schema for user management.
- Create REST API design for user authentication.
- Review and approve wireframes.
- Update designs based on feedback.

3. Development
Objective: Implement functionality and fix bugs.
Issues:

- Develop login functionality.
- Implement user authentication API.
- Develop password recovery feature.
- Fix bug in login form validation.
- Integrate database schema for user management.
- Add session management for user login.
- Refactor code for API performance.

4. Testing
Objective: Perform functional, integration, and regression testing.
Issues:

- Write test cases for login functionality.
- Perform integration testing for user authentication API.
- Conduct regression testing for the entire application.
- Fix identified bugs in password recovery feature.
- Retest user management workflow.
- Verify performance of the login system under load.

5. Release
Objective: Prepare the system for production deployment.
Issues:

- Create deployment scripts for staging environment.
- Perform smoke testing on staging.
- Generate release notes for the new version.
- Deploy application to production.
- Monitor production for post-release issues.
- Fix any high-priority issues found in production.
- Archive sprint and document retrospective findings.

Example Workflow Implementation

1. Requirement Gathering: Issues 1–5 move from Backlog → To Do and then to Done once documented.
2. Design: Issues 6–10 transition from Backlog → To Do → In Progress → Done.
3. Development: Issues 11–17 follow Backlog → To Do → In Progress → Code Review → Done.
4. Testing: Issues 18–23 follow To Do → In Progress → Testing → Done.
5. Release: Issues 24–30 progress through Backlog → To Do → In Progress → Done.

Tips for Managing This Workflow in JIRA

<u>Custom Workflow:</u>
Create custom statuses like Blocked and Code Review to reflect real-life scenarios.

Automation:
Example: Automatically transition issues to Code Review when the status changes to "Ready for Review."

Reports:
Use Burndown Charts to track sprint progress and Cumulative Flow Diagrams to visualize task distribution.

Automation Rules for the Workflow

1. Automatically Transition to "Blocked"
    - Trigger: An issue is updated, and a blocking condition is met.
    - Condition: Issue priority is "High" or labels include "Blocked."
    - Action: Transition issue to "Blocked."

2. Reassign to Original Developer When Unblocked
    - Trigger: Issue transitioned from "Blocked" to "In Progress."
    - Condition: Previous assignee exists.
    - Action: Assign the issue to the previous assignee.

3. Notify Reviewers for Code Review
    - Trigger: Issue transitioned to "Code Review."
    - Condition: Issue type is "Bug" or "Feature."
    - Action: Send a notification to the "Code Reviewers" group.

4. Move to "Done" After QA Approval
    - Trigger: Issue transitioned to "Testing."
    - Condition: QA field is set to "Approved."
    - Action: Transition the issue to "Done."

5. Alert for Overdue Issues in "In Progress"
    - Trigger: Scheduled (daily at 9 AM).
    - Condition: Due date < today AND status = "In Progress."
    - Action: Send an email to the assignee and project lead.