Jira Process Standardization Guidelines

*Designed for Yuzee - EdTech Platform | Implemented Across 5 Teams*  
*Author: Farrukh Nadeem Role: Project Manager & Data Workflow Architect*

1. Board Structure & Philosophy

**Objective:** *"Replace tribal knowledge with data-driven workflows"*

Column Logic

Diagram



Key Innovations:

* **"Reassigned" vs "Assigned"** → Track rework separately
* **"Untestable" status** → Flag environment issues early
* **"New RGT" label** → Isolate regression bugs

2. Status Transition Matrix (Partial)

*(Full version included 28 controlled transitions)*

| **From → To** | **Allowed For** | **Rules & Triggers** | **Data Captured** |
| --- | --- | --- | --- |
| To Do → Next in Line | Manager | Pre-filter tasks for upcoming sprints | Priority, Module |
| Dependency → Assigned | Manager | Dependency resolved + confirmed via comment | Resolution timestamp |
| In Progress → Paused | Developer | Must @mention manager with blockage details | Blockage type, ETA |
| PR Merged → Reopened | Tester | Requires failed test case attachment | Failure pattern, Screenshot |

3. Custom Field Taxonomy

Dependency Types

1. Backend: When the task is blocked due to pending work from backend be it new work or bugs.
2. Requirement Confirmation: When the task is blocked due to requirement not confirmed/clear via Product Team.
3. UI developer: When the task is blocked due to pending work from UI developer be it new work or bugs.
4. Another developer: When the task is blocked due to pending work from another developer be it new work or bugs.
5. Common Component: When the task is blocked because it concerns a common component being worked upon by someone else.
6. Another Section: When the task is blocked due to pending work from another section/component that it depends on, be it new work or bugs.
7. Approval Needed When the task is blocked because it awaits approval from the Administration.

Bug Classification

| **Type** | **Detection Method** | **Example** |
| --- | --- | --- |
| API | Postman/Console errors | 500 error on student\_profile GET |
| Logic | Flowchart deviation | User bypasses age verification |
| Validation | Boundary testing | Future date accepted in "DOB" field |

Labels

## Labels

This field captures the nature of work on any component. These labels or work type can be associated with any component as they are common across the application being built. For e.g., a component “Contact Details” can have a create form, details page, edit popup, triggers etc., just like any other component.

1. Api: When the task involves working on or with the API specifically. For e.g., an API needs to be integrated in order to GET the contact details of an Institute.
2. createForm: When the task involves working on the create forms of any component such as entering personal details via a form.
3. DetailsPage: When the task involves working on the details page used to present the relevant information such as showing the list of ways a person can contact an Institute.
4. EditPopups: When the task involves working on the edit popups of any component.
5. Empty\_state: When the task involves working on the empty state of any component such as showing the page designed specifically when contact details information has not been provided.
6. Listing: When the task involves working on or with the list related to any component for e.g., we have a component called courses. So to display the list of courses being offered we have to work upon such feature.
7. Preview: When the task involves working on displaying the preview page.
8. Triggers: When the task involves working on or with the triggers that are relevant to any component such as working on a button that is used to save the information of a course into the Institute’s profile.

4. Enforcement Mechanisms

**Validation Rules**

1. **Ticket Blocking:**
   * Cannot move Assigned → In Progress without:  
     ✅ Estimated hours logged  
     ✅ Module/Component selected
2. **Dependency Reporting:**
   * Cannot move (any) → Dependency without:  
     ✅ Defining dependency type  
     ✅ Mentioning the responsible team and requirement

Document Appendices

**Sample Ticket**

<https://yuzeeff.atlassian.net/browse/YUZ-9>

*Fields shown: Epic, Dependency Type, Blocked comment, Component*