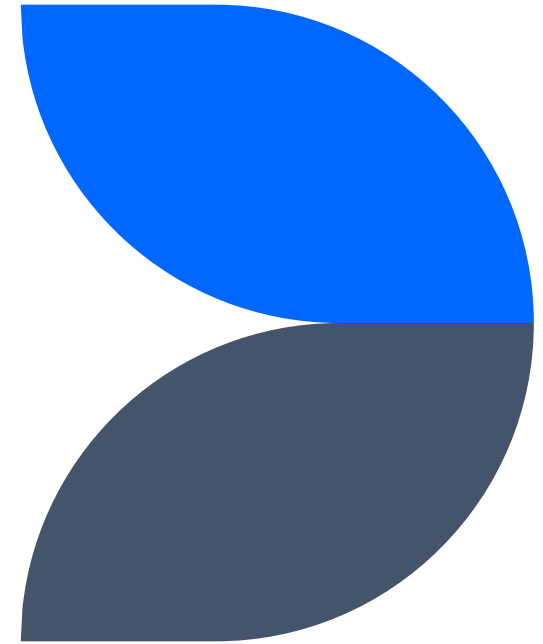
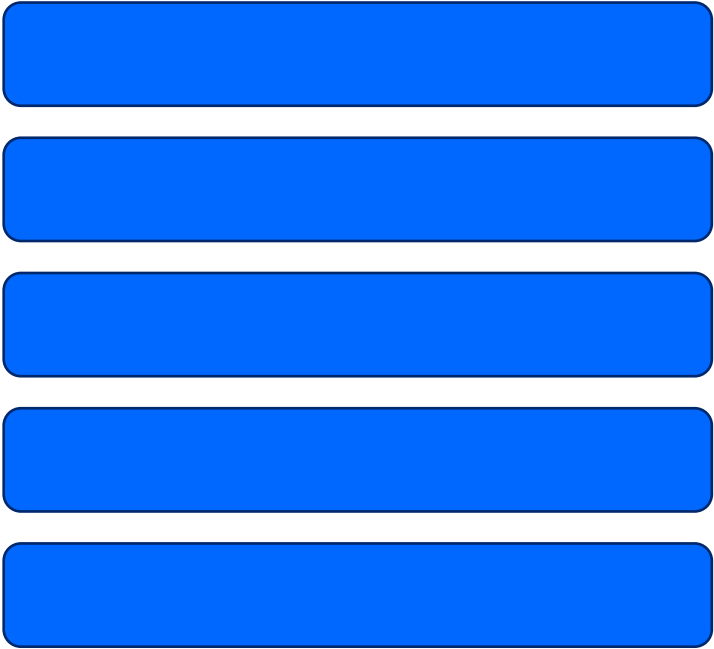


Lab 10:

Custom Workflows and Transition Rules

CSE233 Agile Software Engineering





What Is a Workflow in Jira?

A workflow defines the lifecycle of an issue.

A workflow consists of:

Statuses

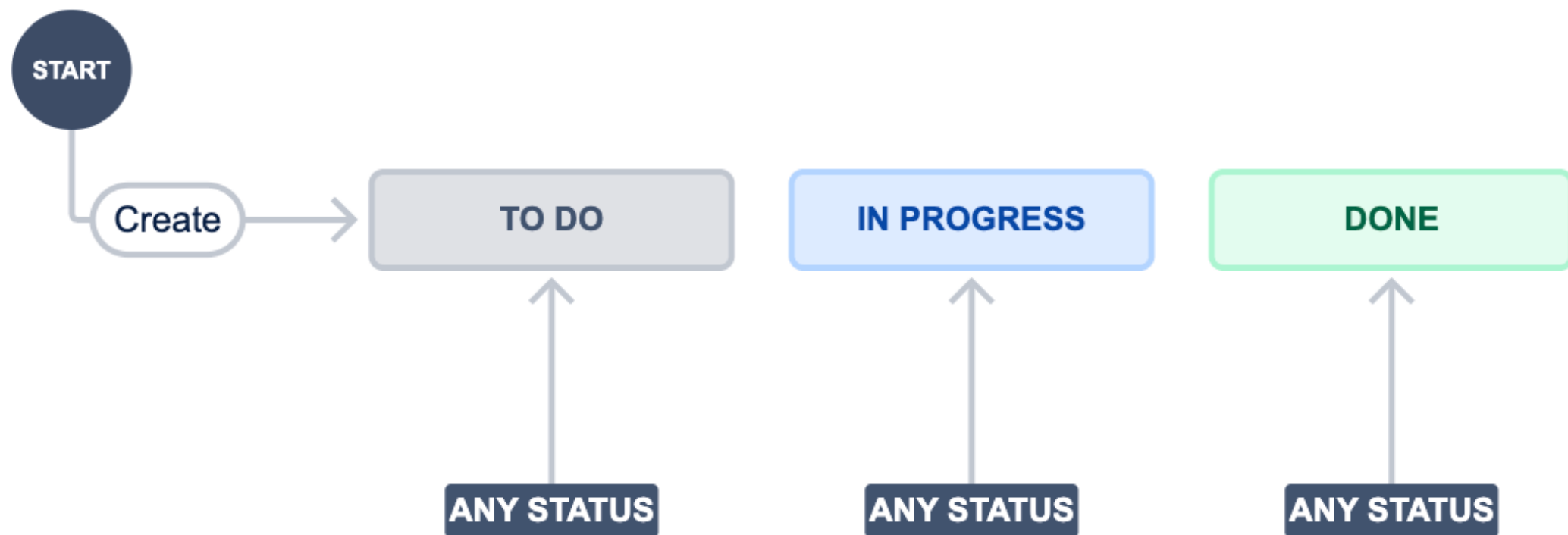
Represent the state of an issue (To Do, In Progress, Done...)

Transitions

How an issue moves between statuses

Rules

Conditions, validators, and post-functions



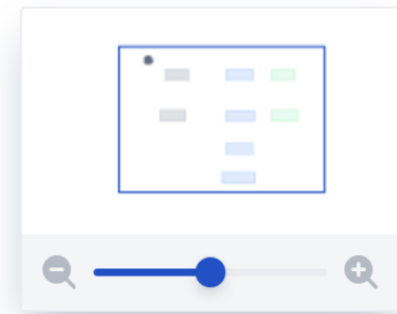
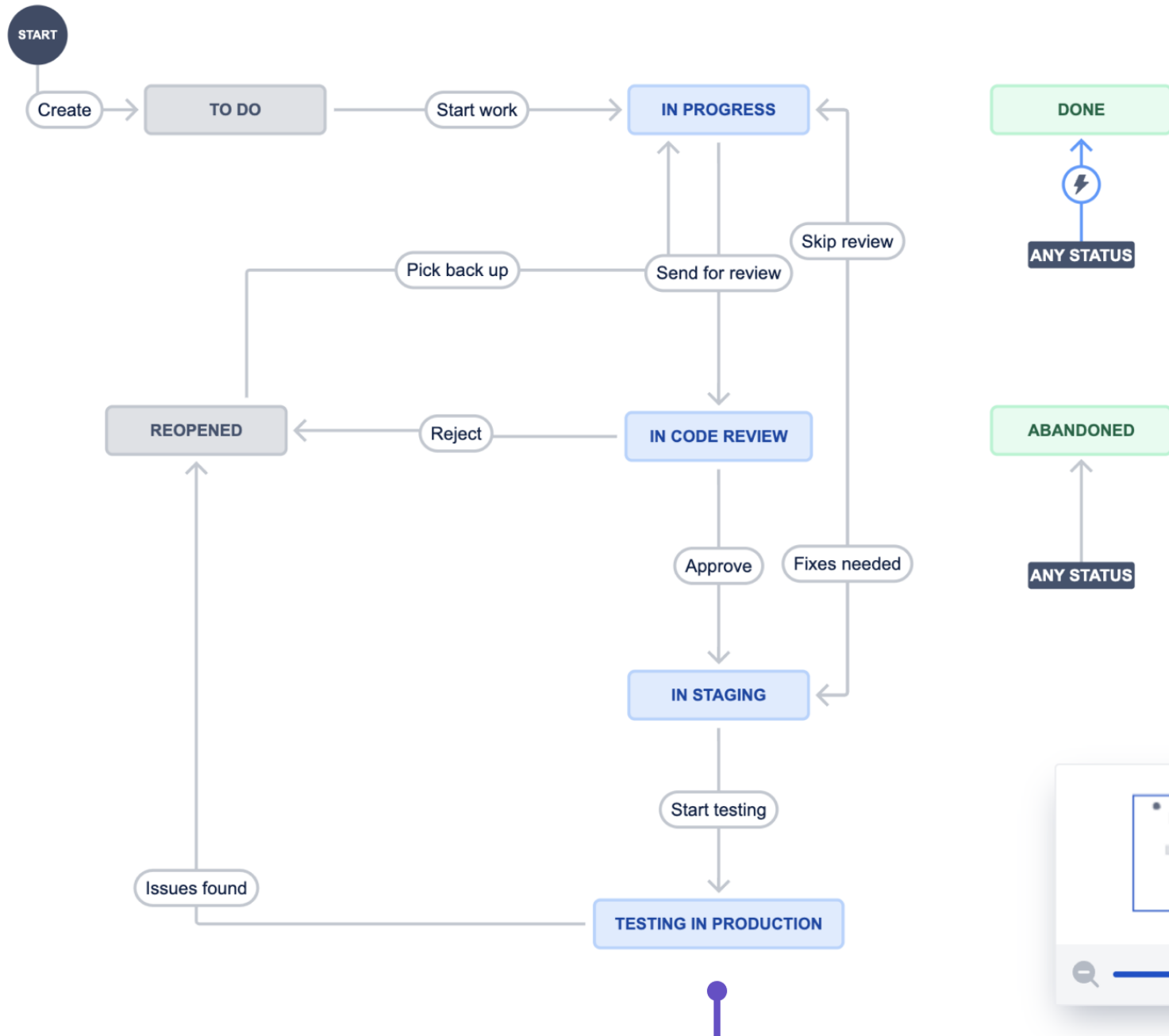
Why Customize Workflows?

Default workflow is very simple:

To Do → In Progress → Done

Teams often need:

- QA verification
- Blocking states
- Approval flows
- Automation on transitions



Status

Statuses represent the different steps of your process.

Each status has:

- **Name**
- **Category** (To Do / In Progress / Done)
- **Meaning for reporting**
(Category affects burn-down charts, cycle time, etc.)

To Do

Work not yet started.

In Progress

Work currently being executed.

Done

Completed work, contributes to velocity.

Transitions

Transitions define **how an issue changes status**.

Each transition has:

- Name (Start work, Move to review)
- Rules (restrictions, validations, actions)

Rules

There are three rule types in Jira Workflows:

1. Restrict Transition (previously Conditions)

Control *who* can execute a transition.

2. Validate Details (previously Validators)

Control *what must be true* before transitioning.

3. Perform Actions (previously Post-Functions)

Execute *automated actions* after transition.

Restrict Transition

A **restrict transition** rule decides who is allowed to perform a transition.

Examples:

- Only the assignee can move issue
- Only users in Developers group
- Only Reporter

Purpose:

Prevent unauthorized or accidental transitions.



Validate Details

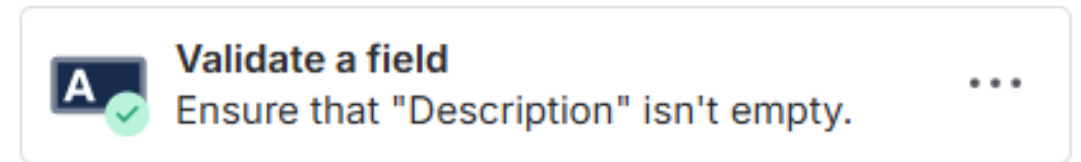
A **validate details** rule checks whether the issue meets requirements before moving.

Examples:

- Description must not be empty
- Issue must have a fix version
- Resolution must be set
- Field must be numeric or non-empty

Purpose:

Avoid bad tickets moving through the system.



Perform Actions

A **perform-actions** rule is executed *after* the transition happens.

Example:

- Set fields automatically (labels, priority, dates, etc.)
- Assign the issue
- Fire an event (notifications)
- Update status/category

Purpose:

Automate repetitive tasks and maintain consistency.



Examples of Custom Workflows

Software Development

To Do → In Progress → Code Review → QA → Done

IT Service Desk

New → Assigned → In Progress → Awaiting Customer → Resolved → Closed

Content Creation

Idea → Draft → Review → Ready → Published

Each team molds Jira to its own process.

Boards VS Workflows



Boards VS Workflows

A board is essentially:

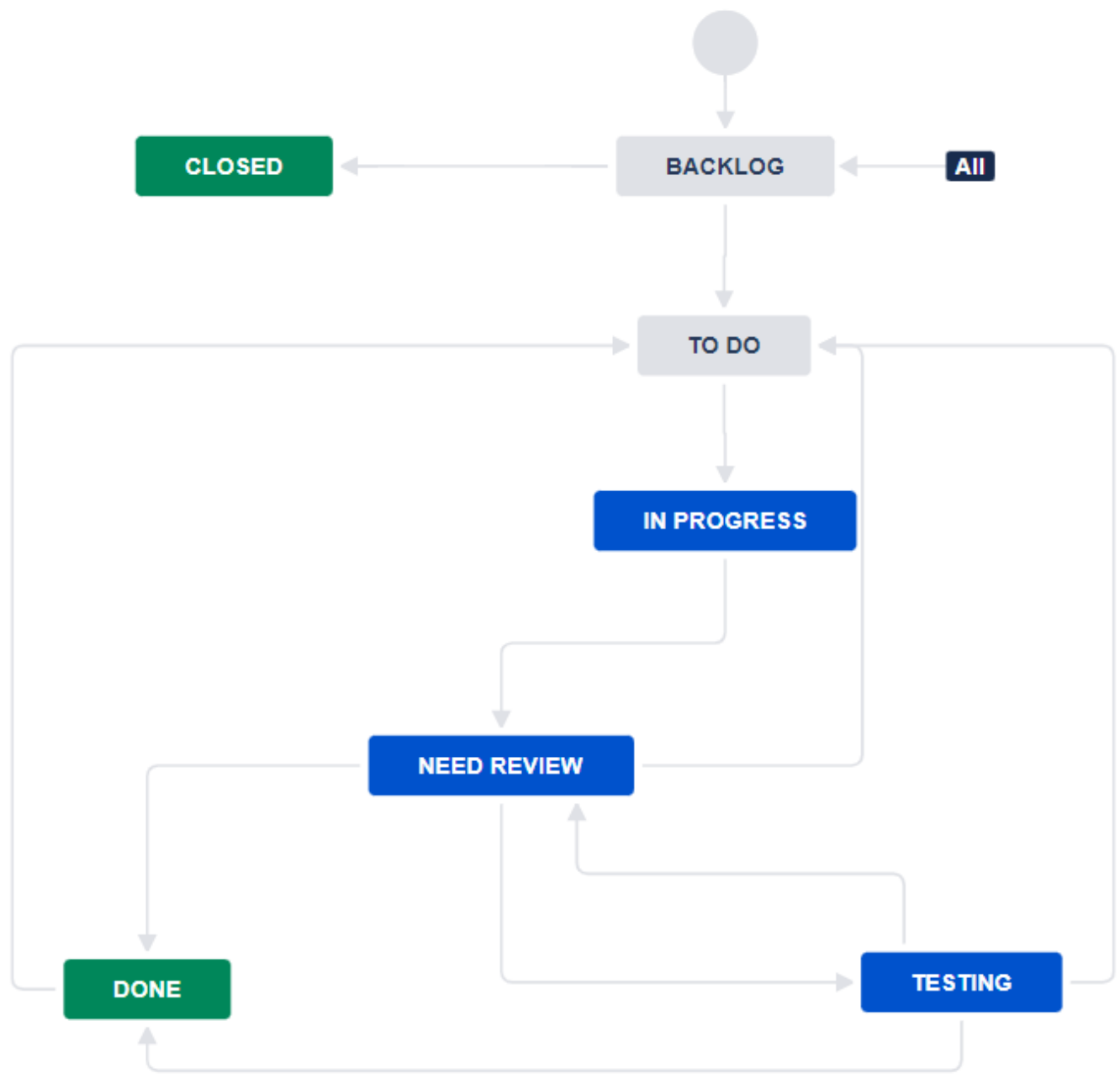
- A **visual layout of statuses** grouped into **columns**
- Powered by the project's **workflow**

This means:

If the workflow changes → the board changes

If statuses are added/removed → columns must be updated

If statuses are unmapped → issues become invisible



Board Columns

Each board column corresponds to **one or more workflow statuses**.

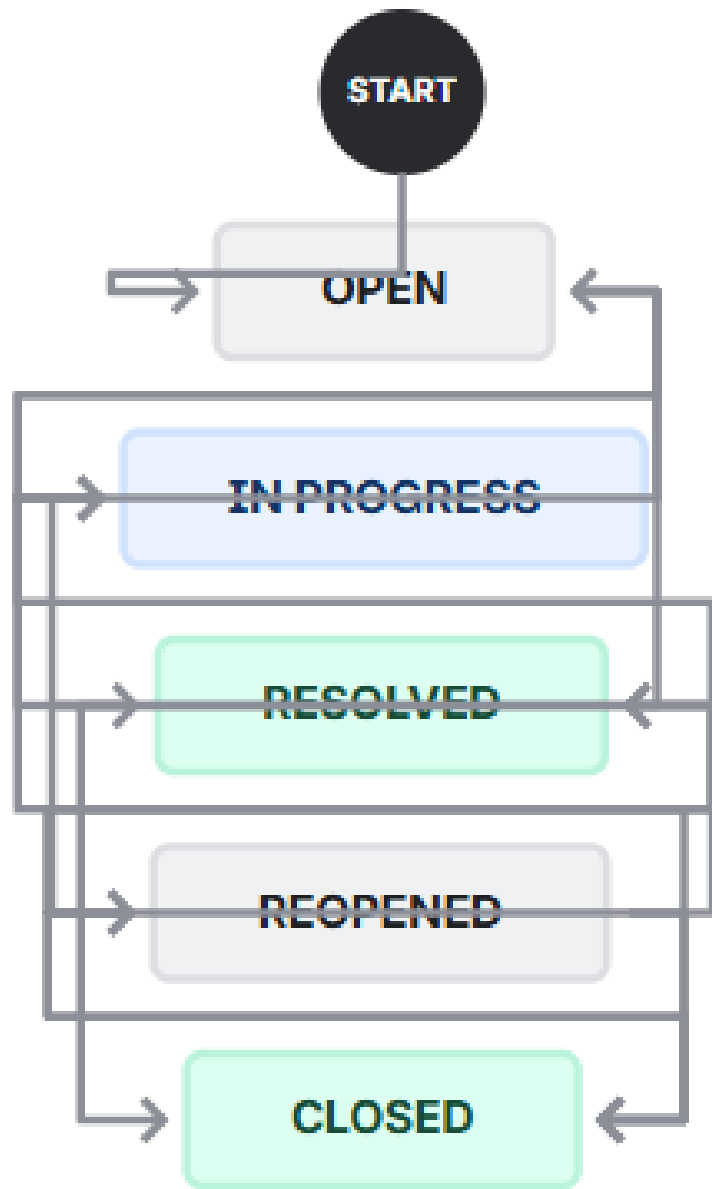
Example column → statuses:

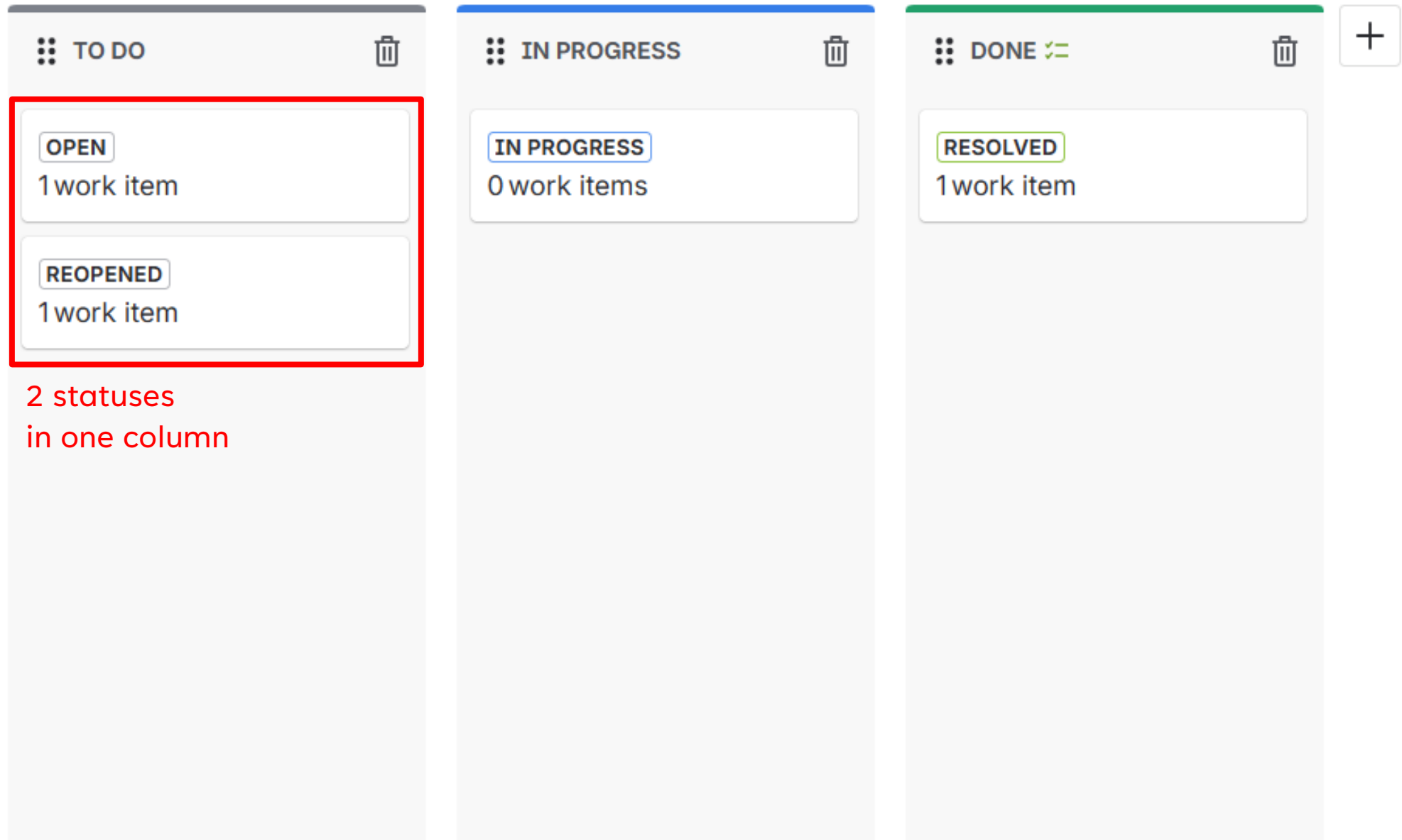
In Progress Column

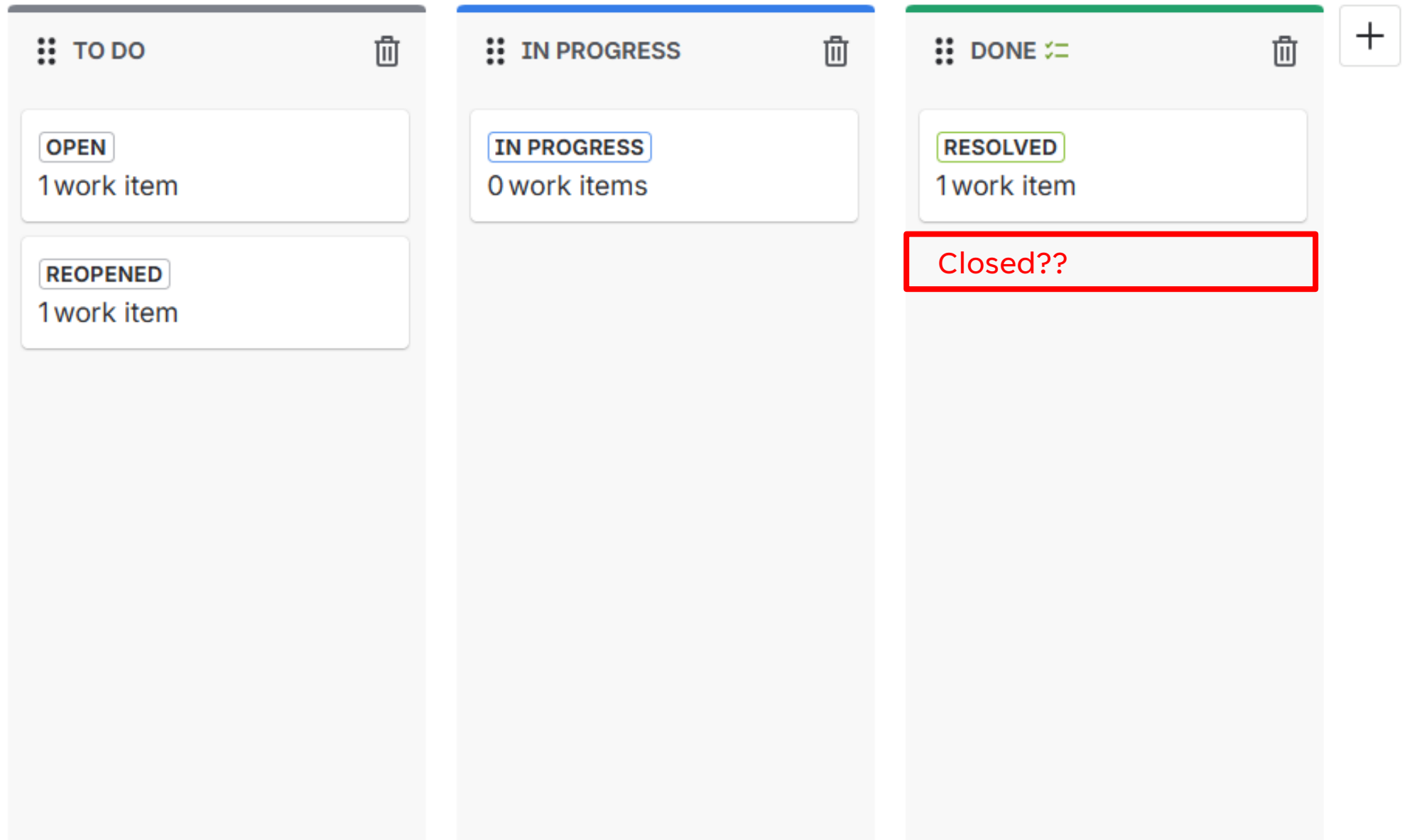
Contains → In Progress, Implementing, Reviewing

Done Column

Contains → Done, Closed, Released (all Done category)







Unmapped Statuses

When a workflow is modified (new status added):

The new status appears under “**Unmapped Statuses**” in Board Settings.

Map statuses to columns i

Unmapped statuses

Work items with these statuses won't be visible.

No work items

CLOSED

0 work items

TO DO



OPEN

1 work item

REOPENED

1 work item

IN PROGRESS



IN PROGRESS

0 work items

DONE ✓



RESOLVED

1 work item



Board Does Not Define the Workflow

Common misconception: “Changing columns changes the workflow.”

Boards are only a *visualization*.

Workflow is defined in: **Project Settings** → **Workflows**

Board is defined in: **Board Settings** → **Columns**

Changing the board does NOT:

- ✗ Add a status to workflow
- ✗ Add transitions
- ✗ Change rules

It only changes how statuses are displayed.



Thank you

Eng. Abdelrahman Salah