

1 Customizing Jira Workflows

To succeed here, you should have



- End-user experience with Jira
- Familiarity with projects, issues, issue types, boards
- Basic understanding of schemes, permissions, and notifications
- Basic familiarity with Jira Service Desk

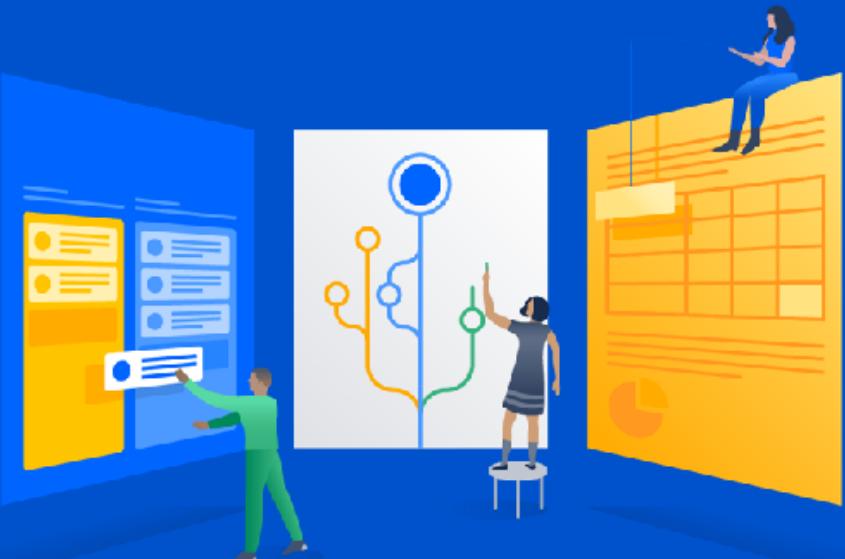


What Will You Learn Here?



- Workflow concepts and terminology
- How to configure transitions to meet complex business requirements
- Where use of Marketplace apps may be appropriate
- How workflows are edited and by whom
- Best practices and troubleshooting





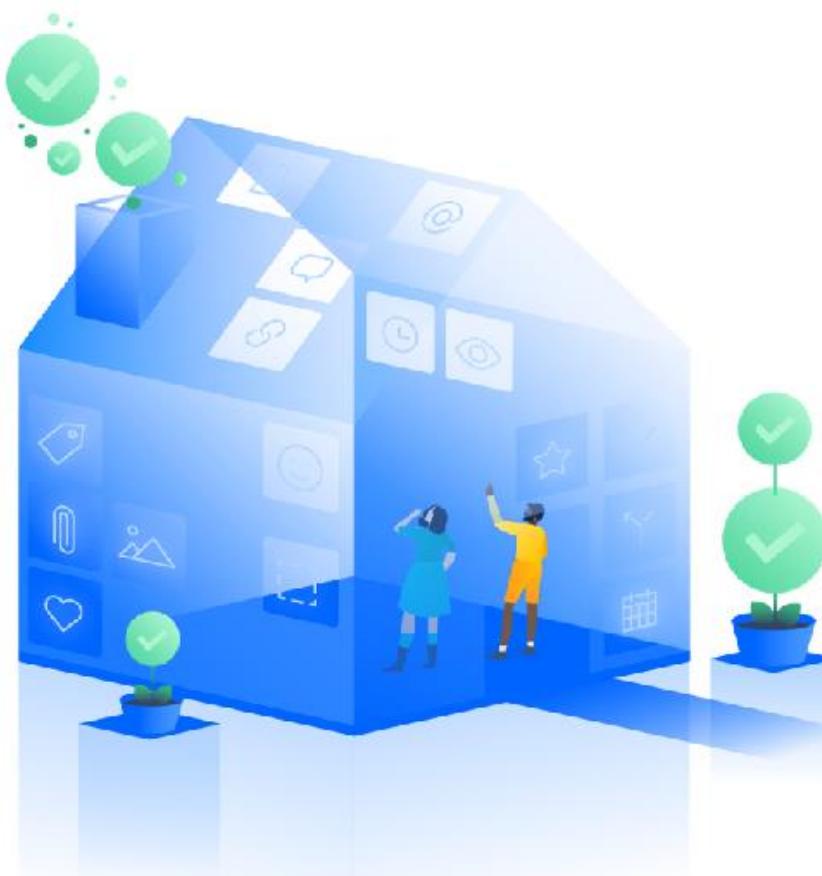
Workflow Basics

- Default Workflows & Customizing
- Configuring Transitions
- Advanced Examples
- Service Desk Approvals & Automation
- Editing and Testing
- Common Problems & Troubleshooting
- Best Practices

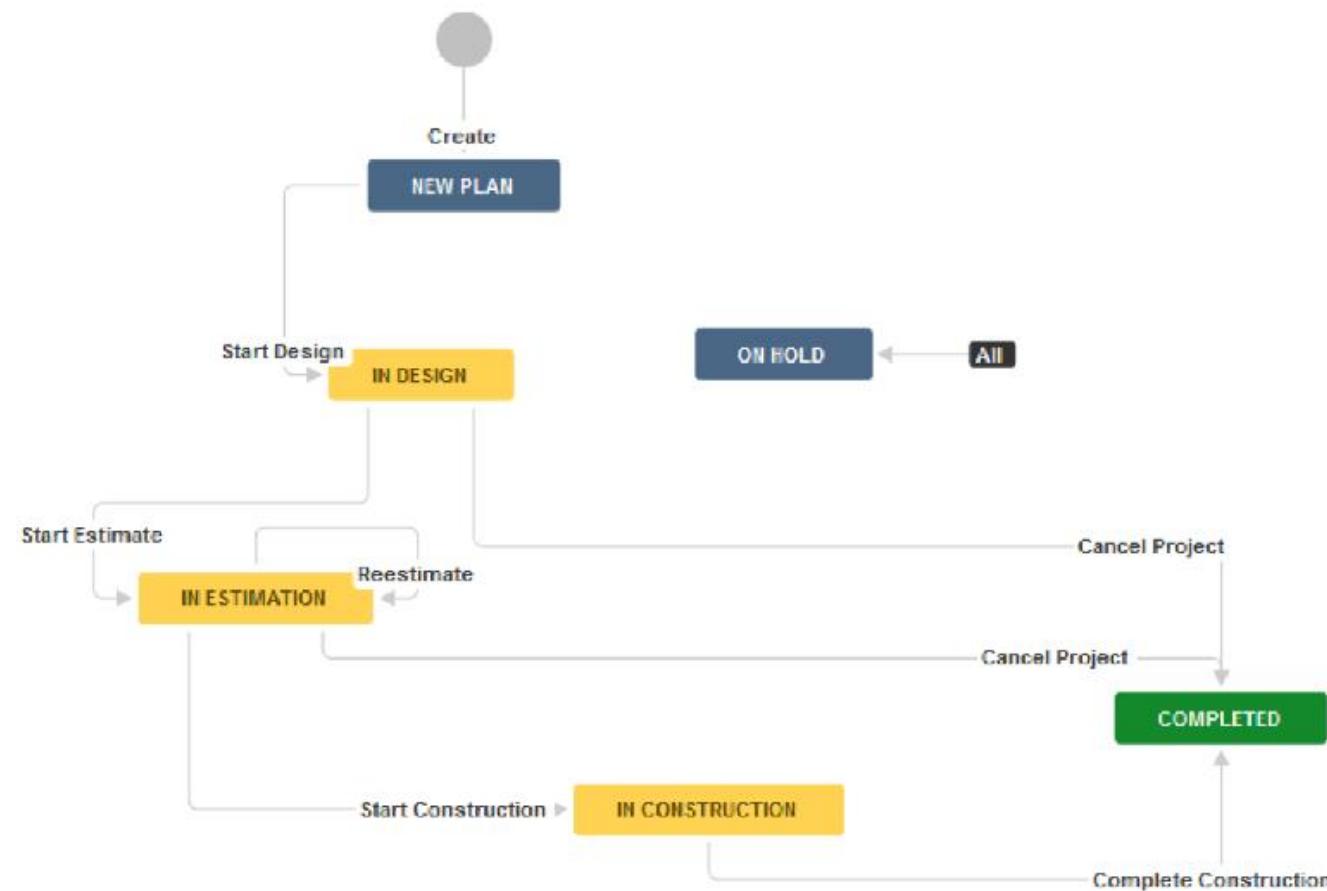


What Is a Workflow?

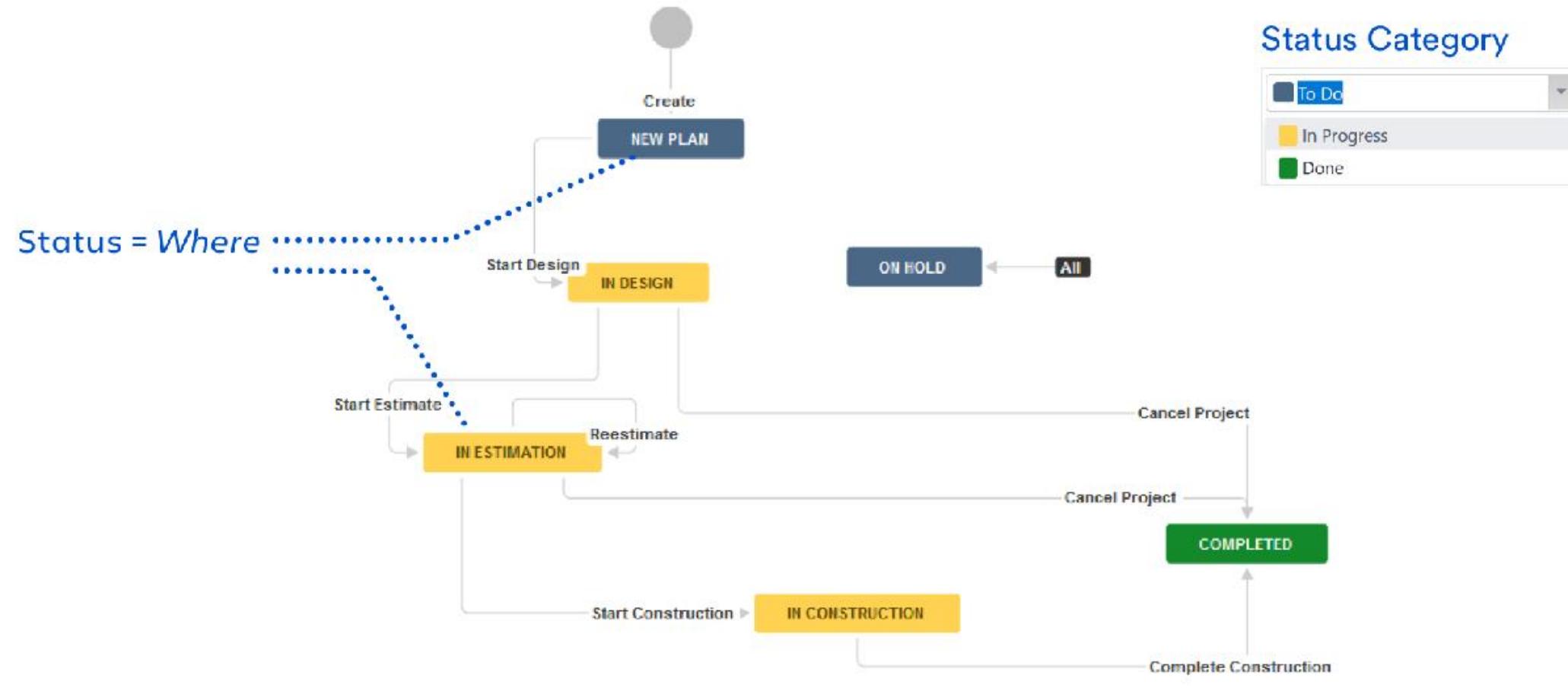
- Process of completing tasks
- Defines the stages from creation to completion
- Different workflows for different types of tasks



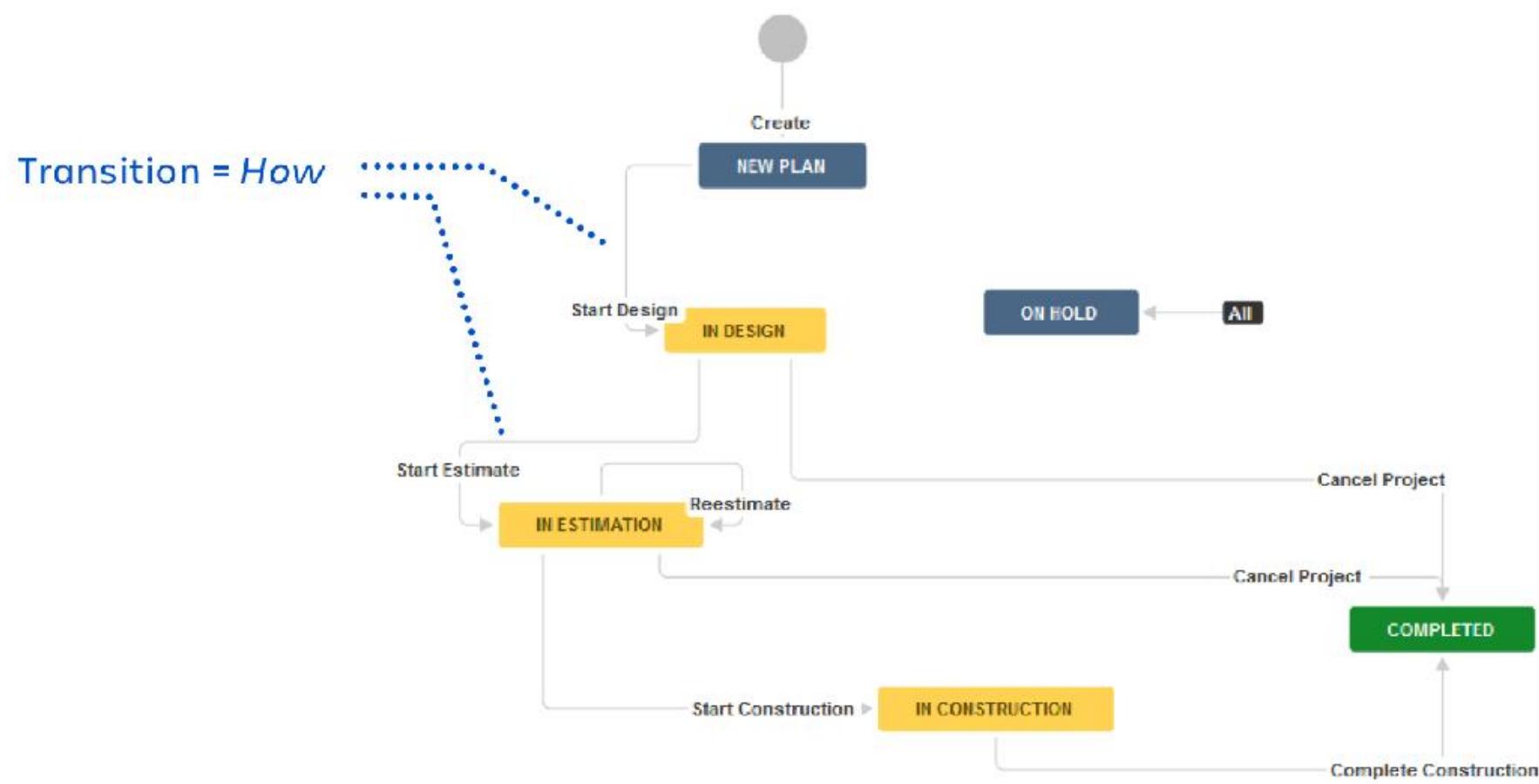
House-Building Workflow



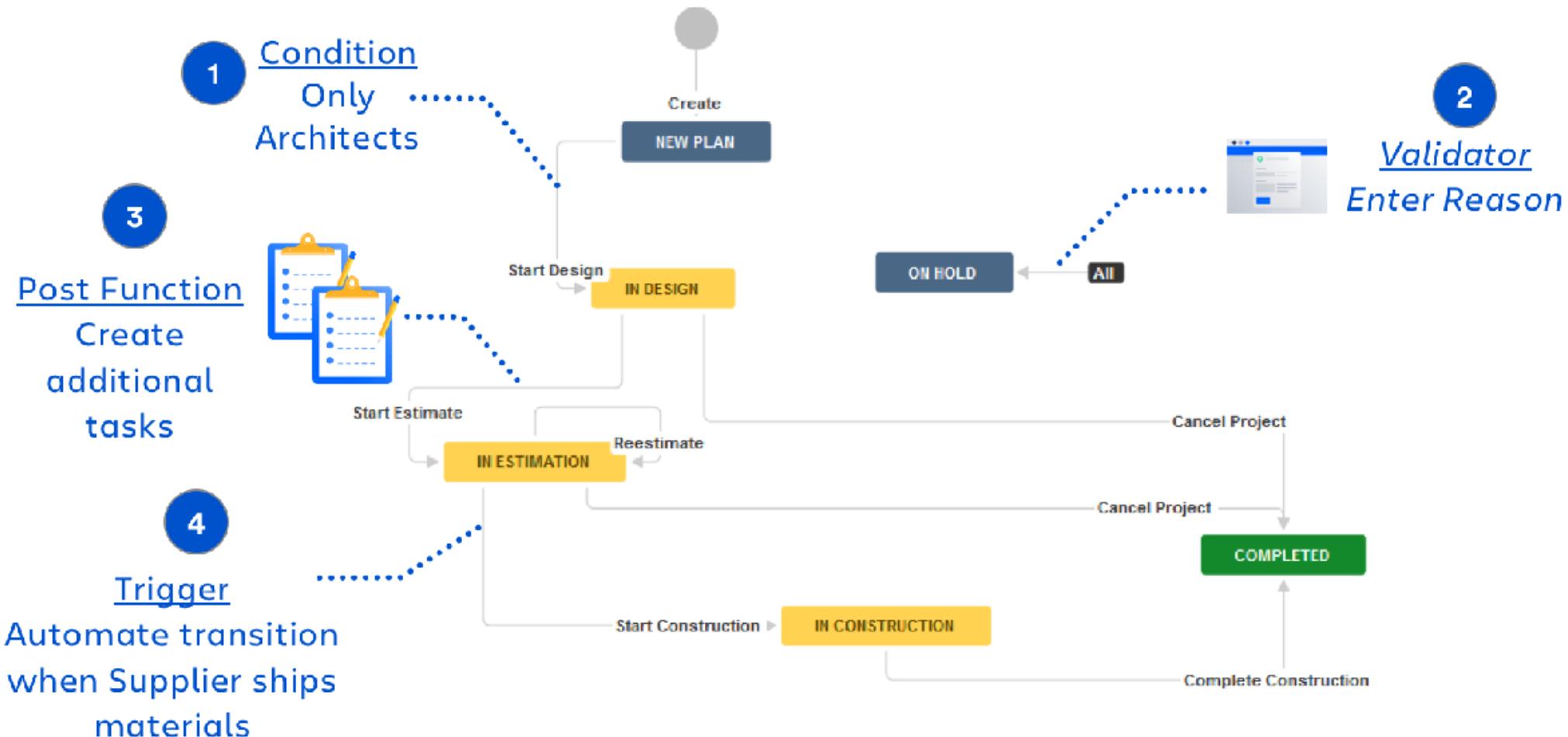
Statuses



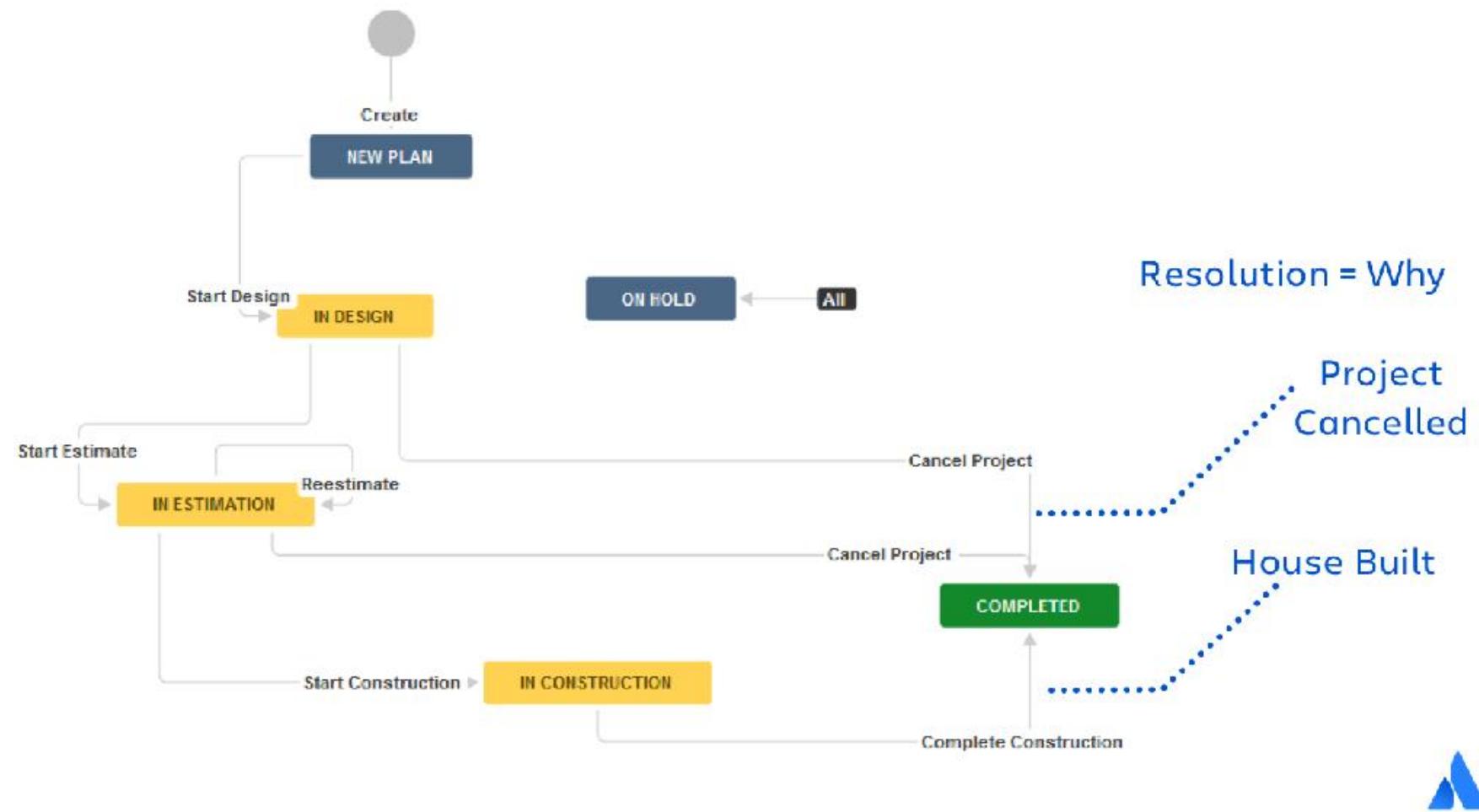
Transitions



Parts of a Transition

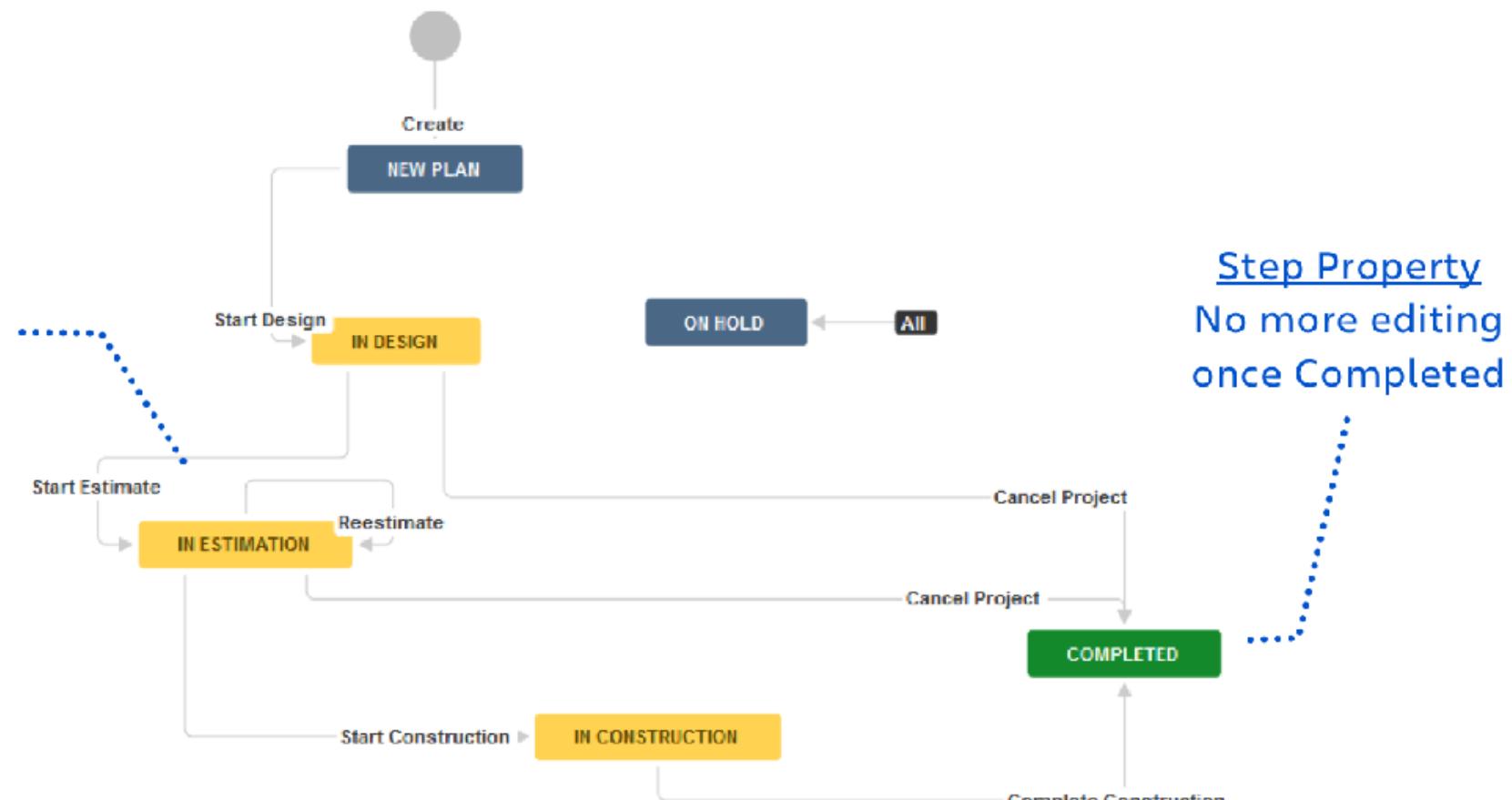


Resolutions

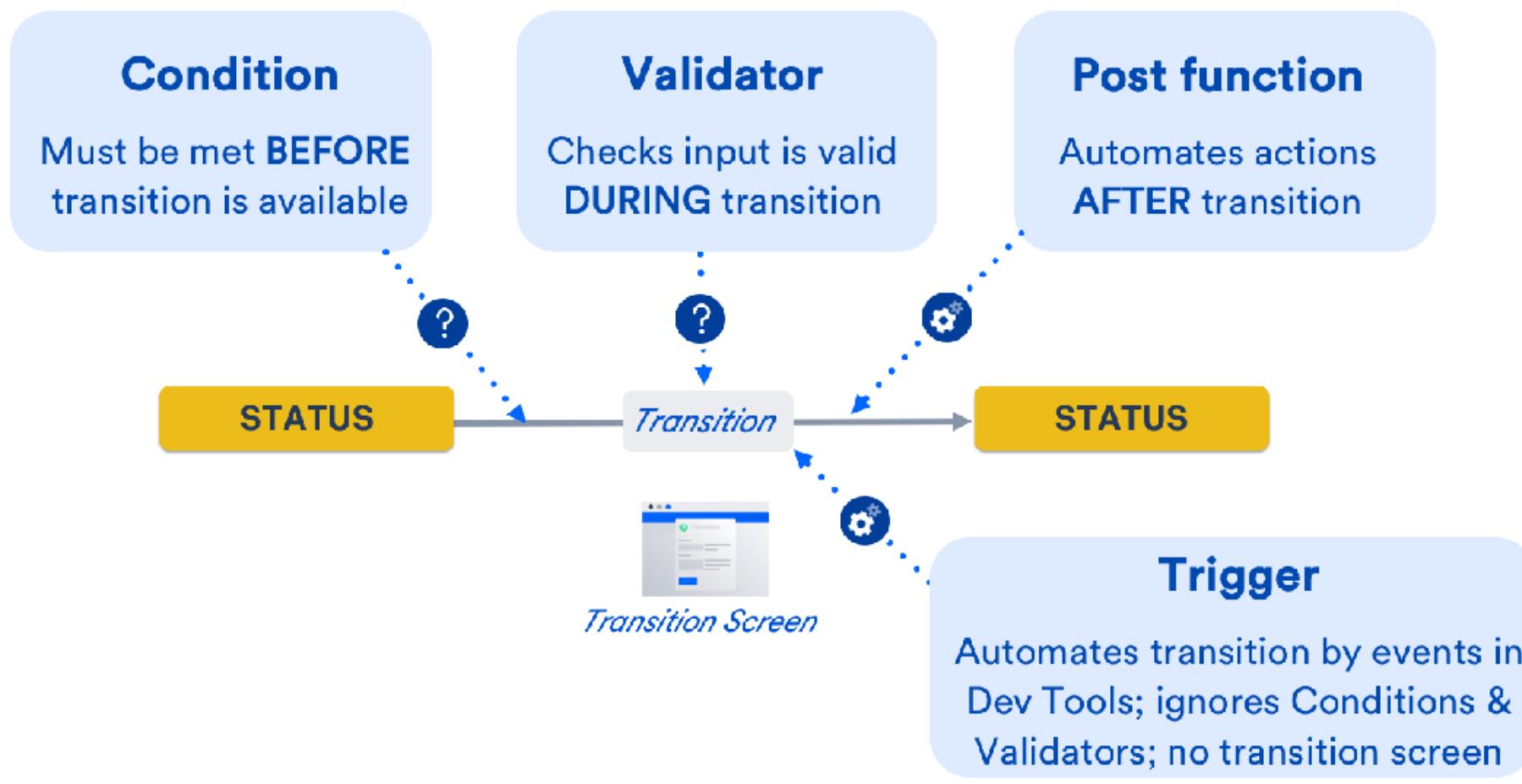


Workflow Properties

Transition Property
Translate buttons
for foreign
companies



Aspects of a Transition's Behavior



Workflow Permissions



To use workflows users need:

- **Browse Projects** permission
- **Transition Issues** permission
- Any other permissions (or group / project role) required by conditions and validators in a workflow transition
- Jira Software Application access (to transition in boards)
- Jira Service Desk Application access (to transition in Service Desk projects)



Workflow Schemes

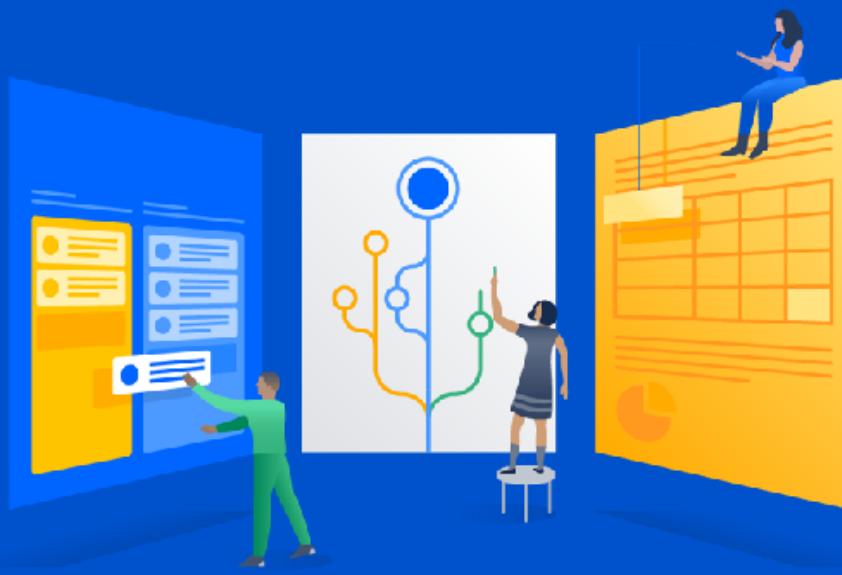
Workflow schemes map issue types to workflows, and are associated with one or more projects.

Name	Projects	Issue Type	Workflow
Dev Workflow Scheme	• Space Programs 	 Unassigned Types  Task  Story  Bug  Sub-task	→ Dev Workflow → Task Workflow → Story Workflow → Bug Workflow → Task Workflow



An extra layer exists in Service Desk projects, where Request Types are first mapped to Issue Types





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

Editing and Testing

Common Problems & Troubleshooting

Best Practices



Default Workflows from Project Templates

 Jira Core

 Jira Software

 Jira Service Desk



Project management

Plan, track and report on all of your work within a project.



Scrum software development

Agile development with a board, sprints and stories. Connects with source and build tools.



IT Service Desk

Manage incidents, changes, problems and service requests with ITSM workflows.



When created from templates, projects get default workflows which are unique to those projects. They are not shared with other projects unless later changed.



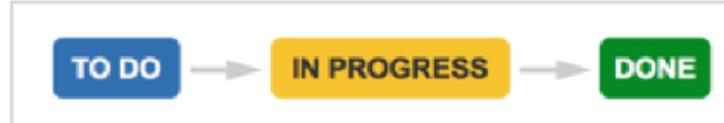
Default Business Workflows

◀ Jira Core

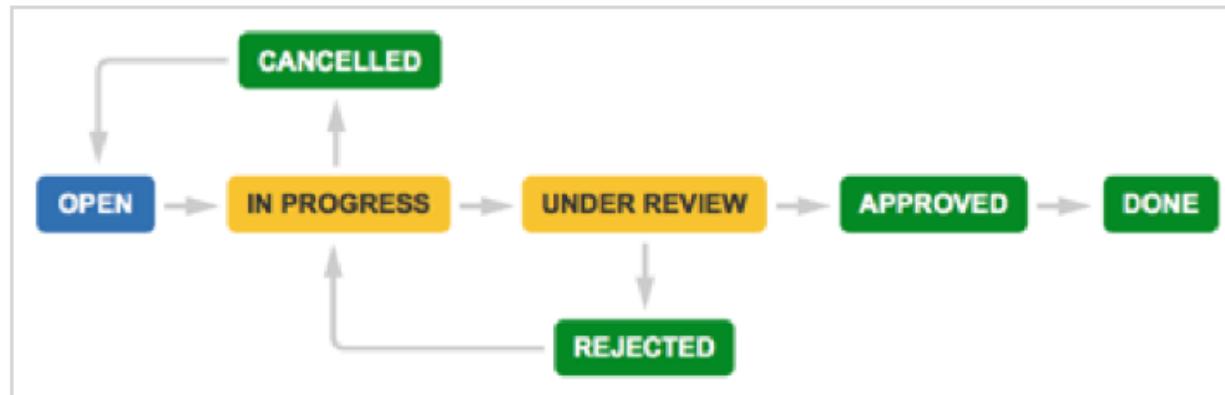
Task management



Project management



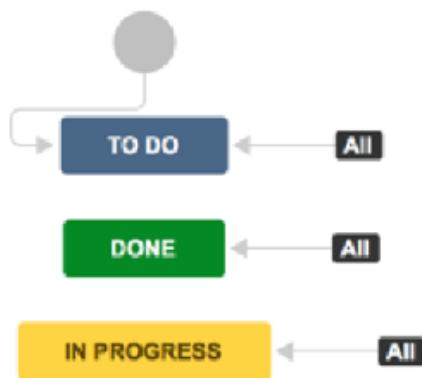
Process management



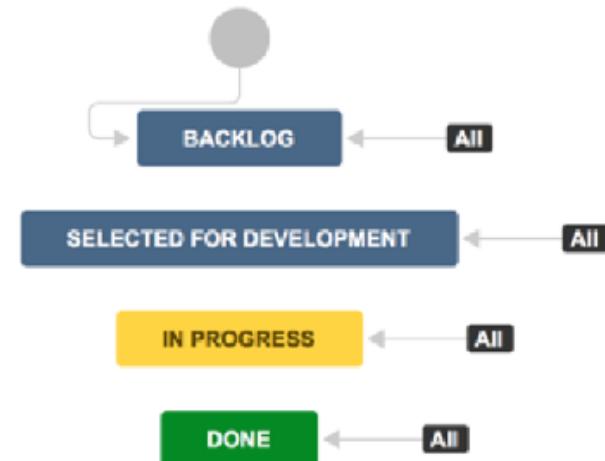
Default Simplified Workflows

Jira Software

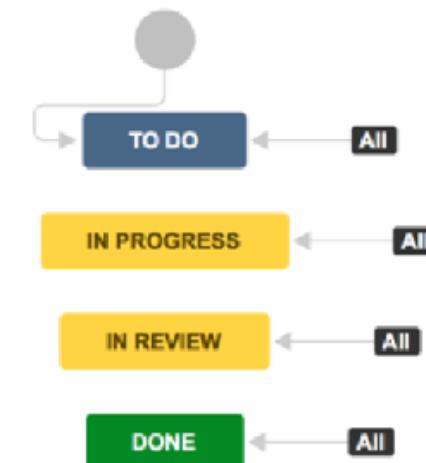
Scrum software development



Kanban software development



Basic software development

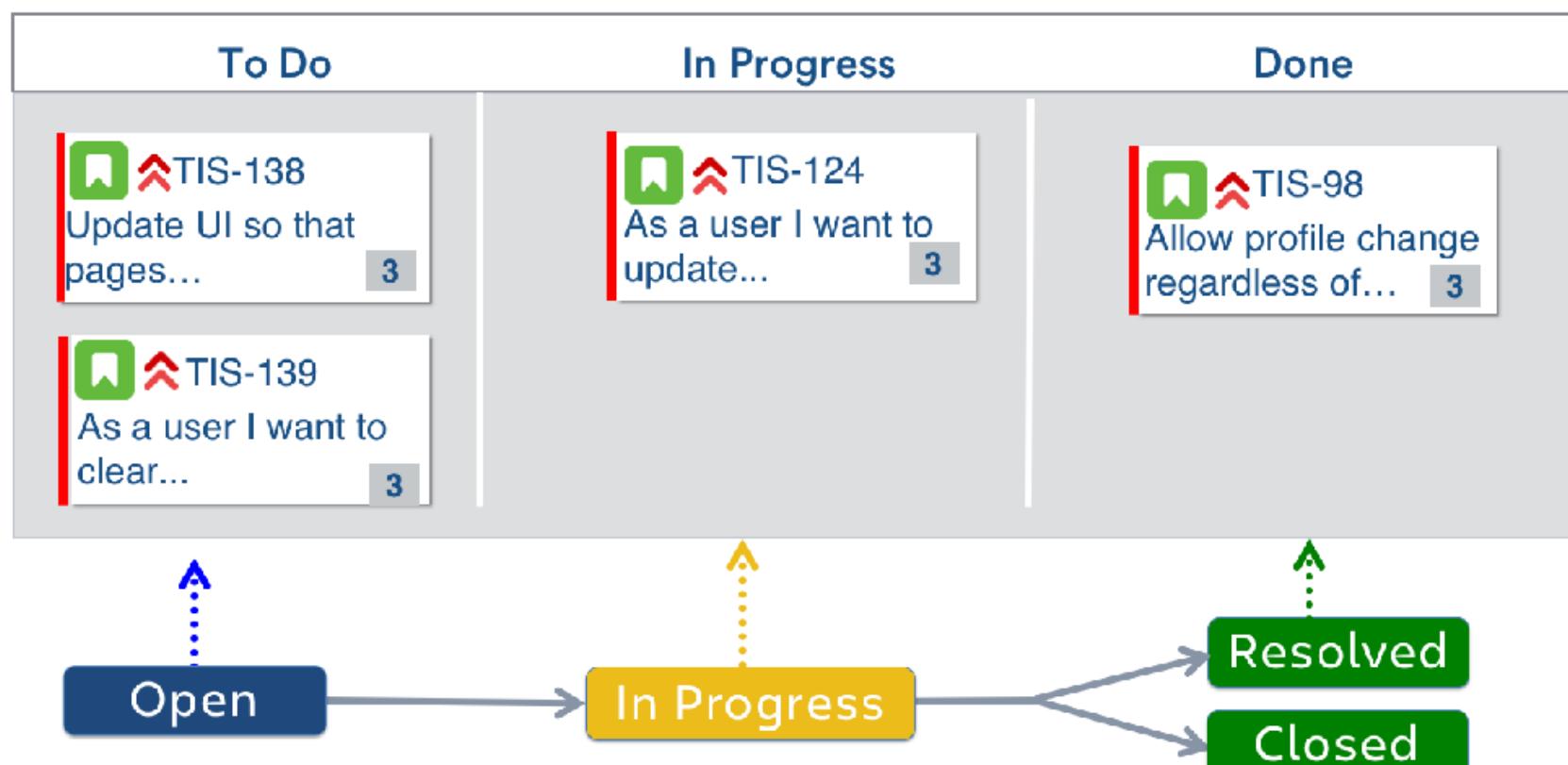


Simplified Workflows contain only Global Transitions



Workflows & Board Columns

Jira Software



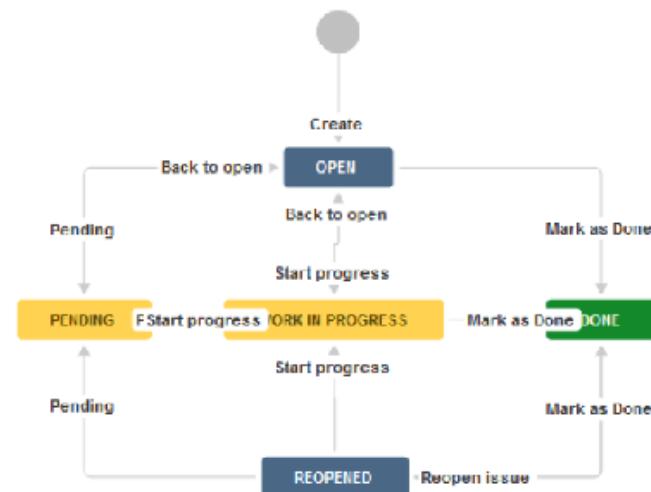
To add a status to a board, the project must be using Simplified Workflow.



Service Desk Default Workflows

⚡ Jira Service Desk

Jira Service Desk default workflow

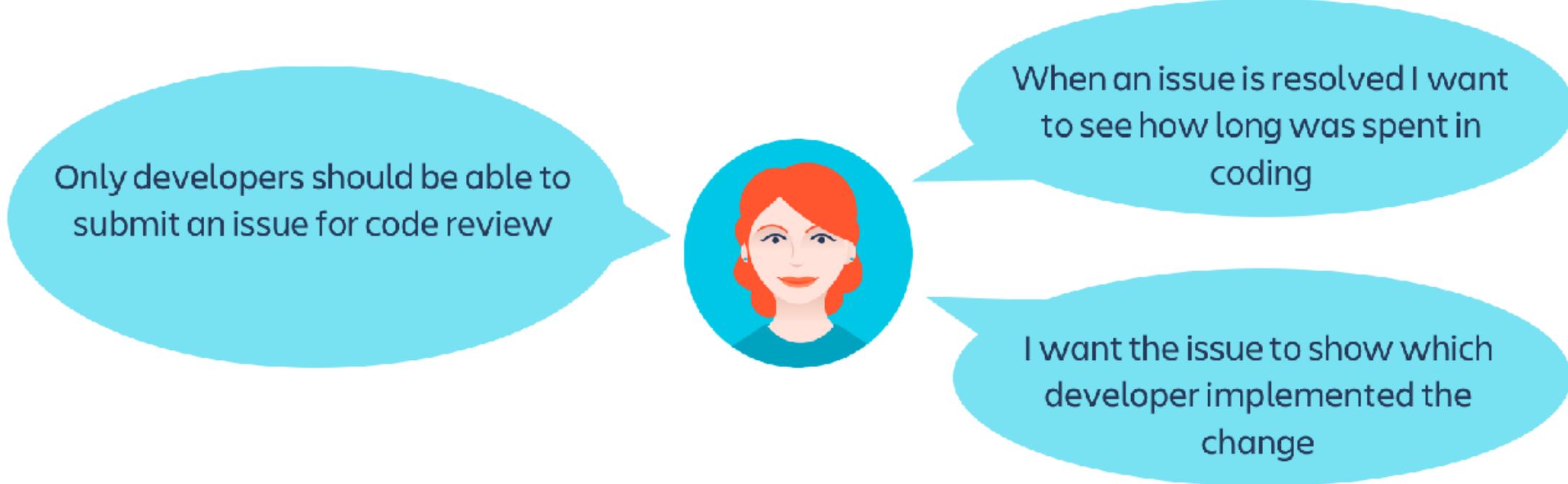


Service Request Fulfilment with Approvals workflow



Why Customize Your Project Workflow?

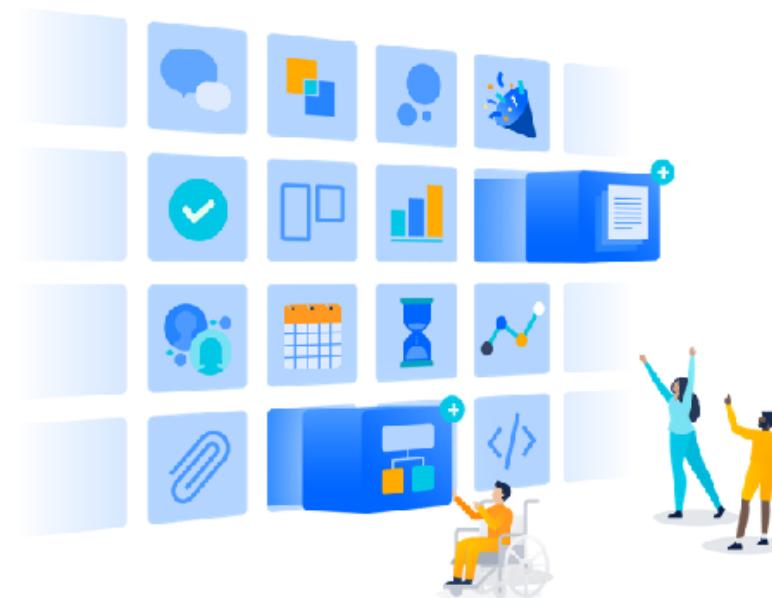
If the default project workflow doesn't meet the needs of your team



Customizing Workflows with Apps

Marketplace apps extend native Jira functionality related to workflows.

- Cloud
- Server
- Data Center



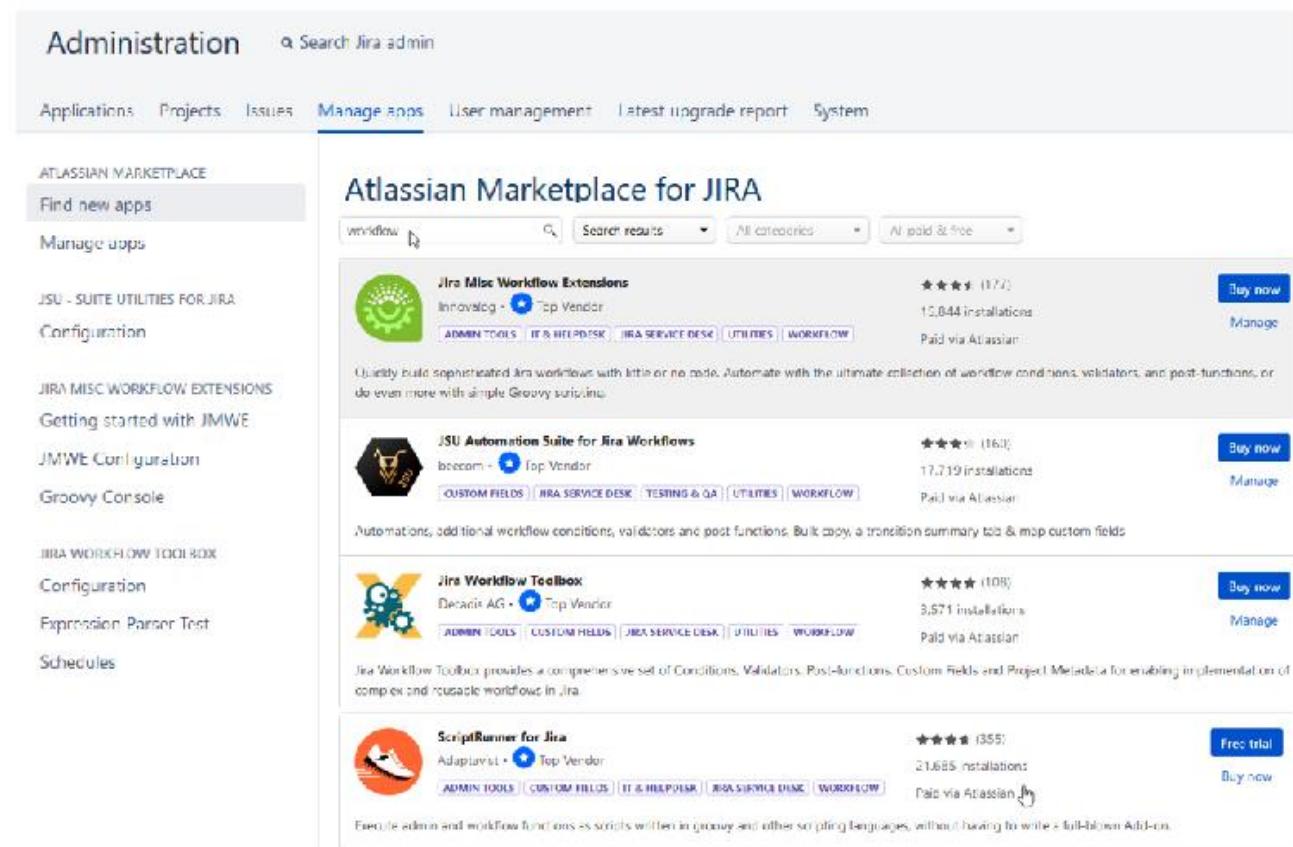
marketplace.atlassian.com



You need Jira System Administrators global permission to install apps.



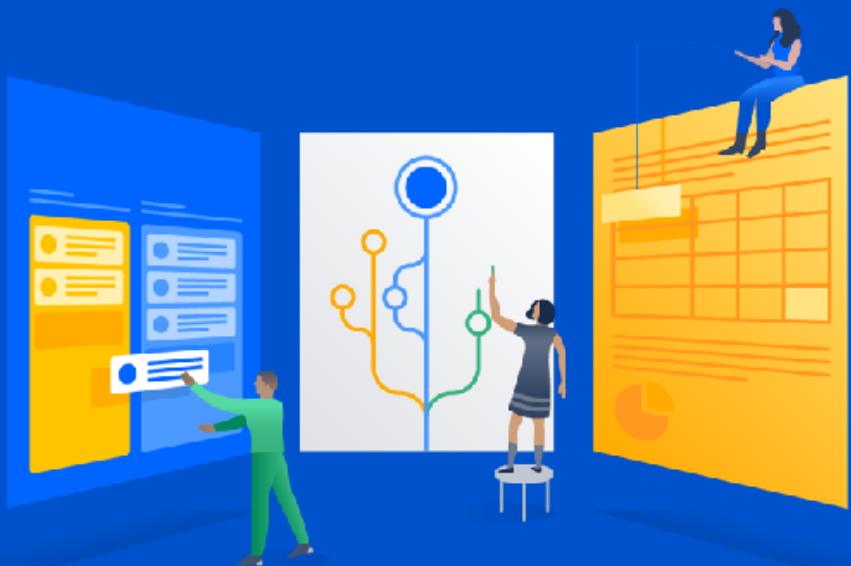
Finding New Apps



The screenshot shows the Atlassian Marketplace for JIRA interface. The top navigation bar includes links for Administration, Applications, Projects, Issues, Manage apps (which is selected), User management, Latest upgrade report, and System. A search bar at the top right contains the placeholder text "Search Jira admin". On the left, there's a sidebar with sections for ATASSIAN MARKETPLACE (Find new apps, Manage apps), JSU - SUITE UTILITIES FOR JIRA (Configuration), JIRA MISC WORKFLOW EXTENSIONS (Getting started with JMWE, JMWE Configuration, Groovy Console), and JIRA WORKFLOW TOOL BOX (Configuration, Expression Parser Test, Schedules). The main content area is titled "Atlassian Marketplace for JIRA" and displays search results for "workflow". The results show four items:

- Jira Misc Workflow Extensions** by Innovacis - Top Vendor (4.5 stars, 177 reviews, 10,844 installations) - Description: Quickly build sophisticated Jira workflows with little or no code. Automate with the ultimate collection of workflow conditions, validators, and post-functions, or do even more with simple Groovy scripting.
- JSU Automation Suite for Jira Workflows** by becom - Top Vendor (4.5 stars, 160 reviews, 17,719 installations) - Description: Automations, additional workflow conditions, validators and post functions. Bulk copy, a transition summary tab & import custom fields.
- Jira Workflow Toolbox** by Decaris AG - Top Vendor (4.5 stars, 108 reviews, 3,571 installations) - Description: Jira Workflow Toolbox provides a comprehensive set of Conditions, Validators, Post-functions, Custom Fields and Project Metadata for enabling implementation of complex and reusable workflows in Jira.
- ScriptRunner for Jira** by Adaptrivs! - Top Vendor (4.5 stars, 355 reviews, 21,685 installations) - Description: Execute admin and workflow functions as scripts written in groovy and other scripting languages, without having to write a full-blown Add-on.





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

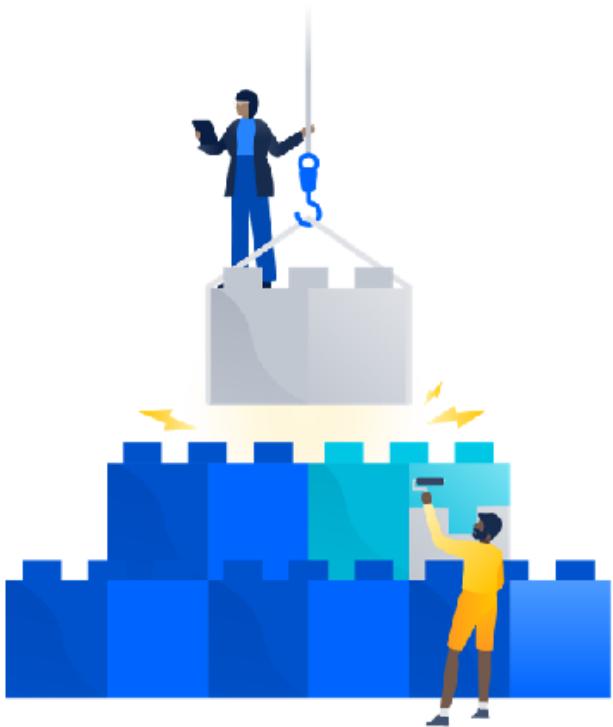
Editing and Testing

Common Problems & Troubleshooting

Best Practices



Configuring Transitions



Conditions

Validators

Post Functions

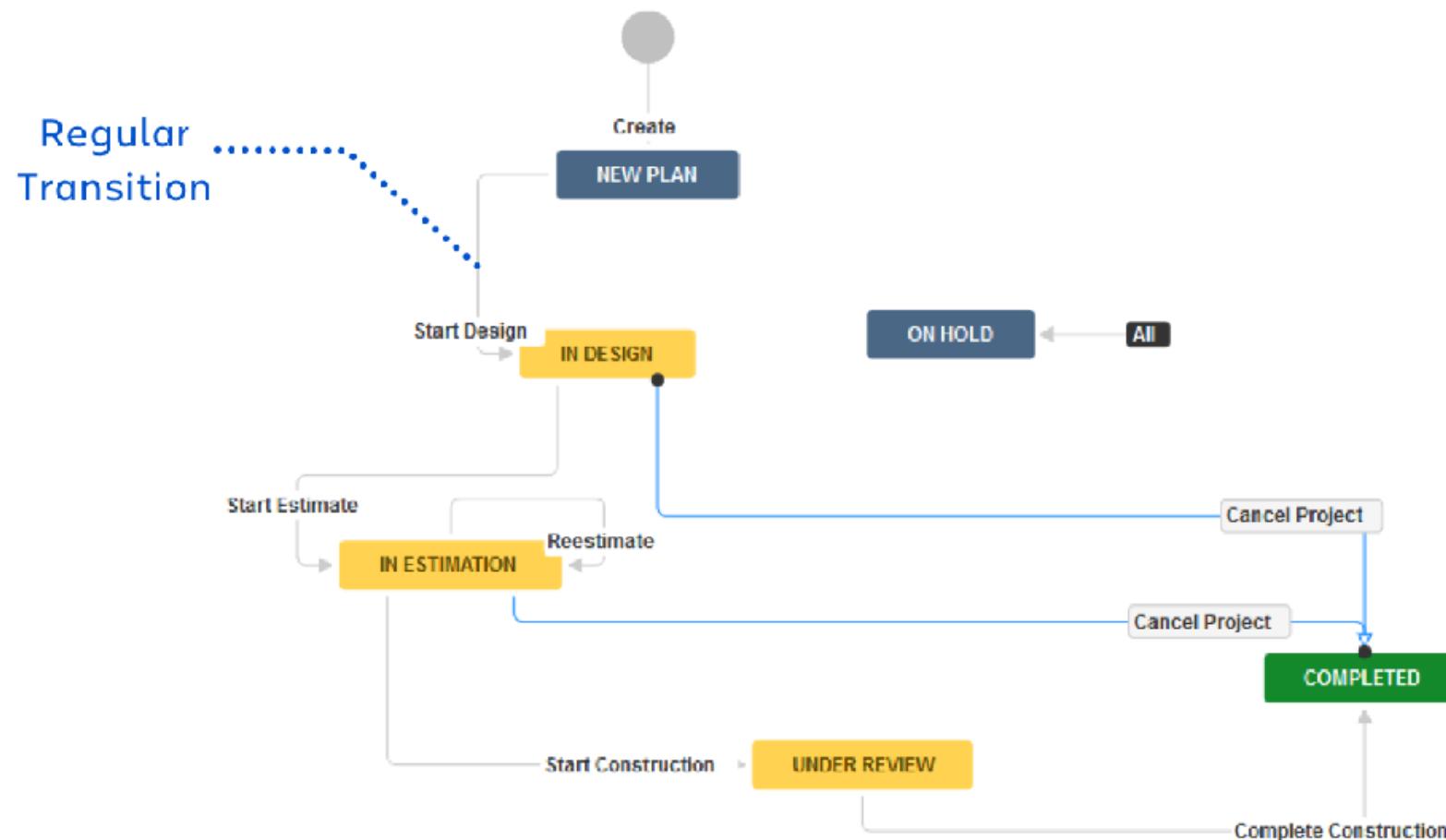
Triggers

Properties

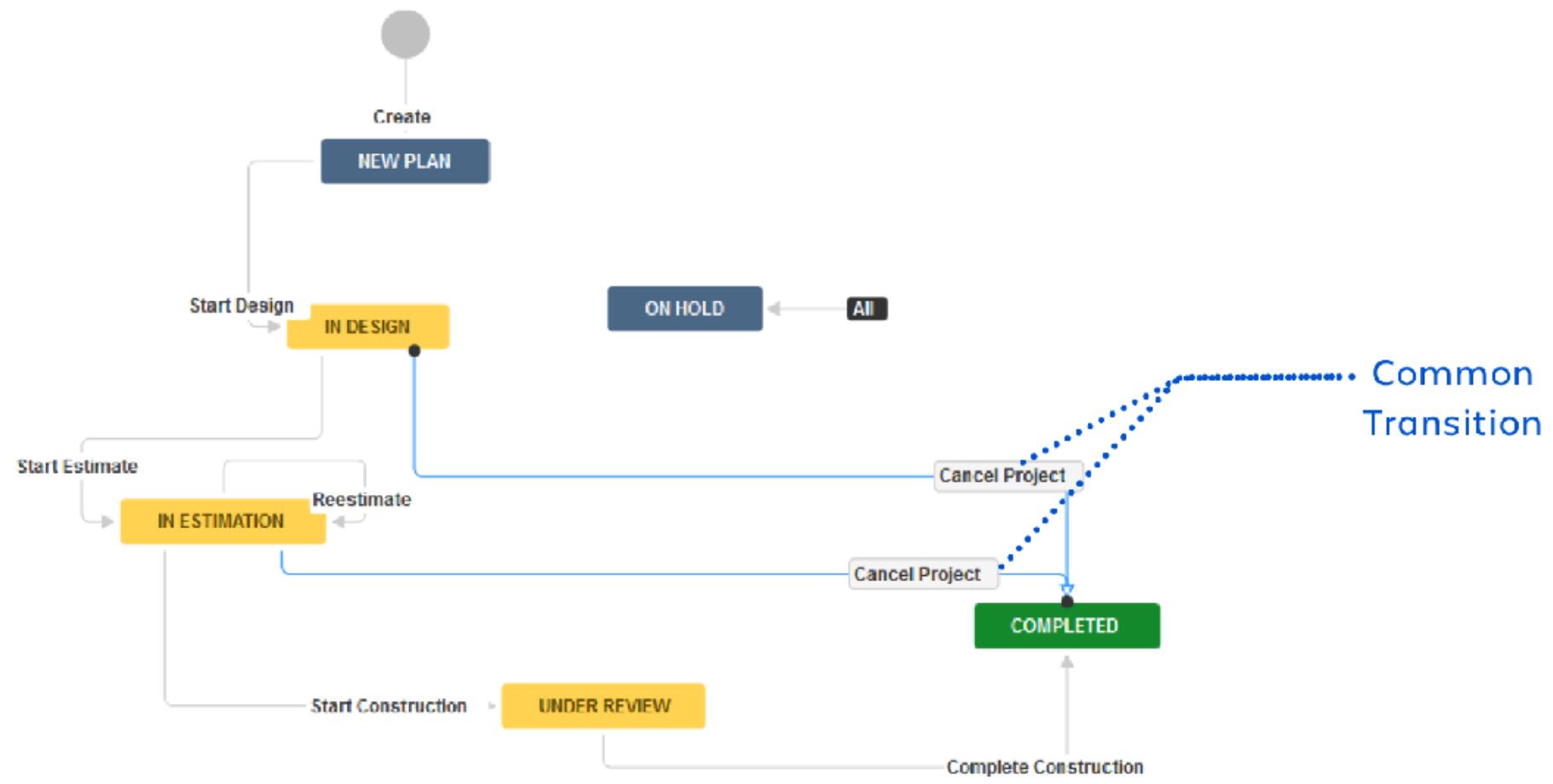
Create Transition



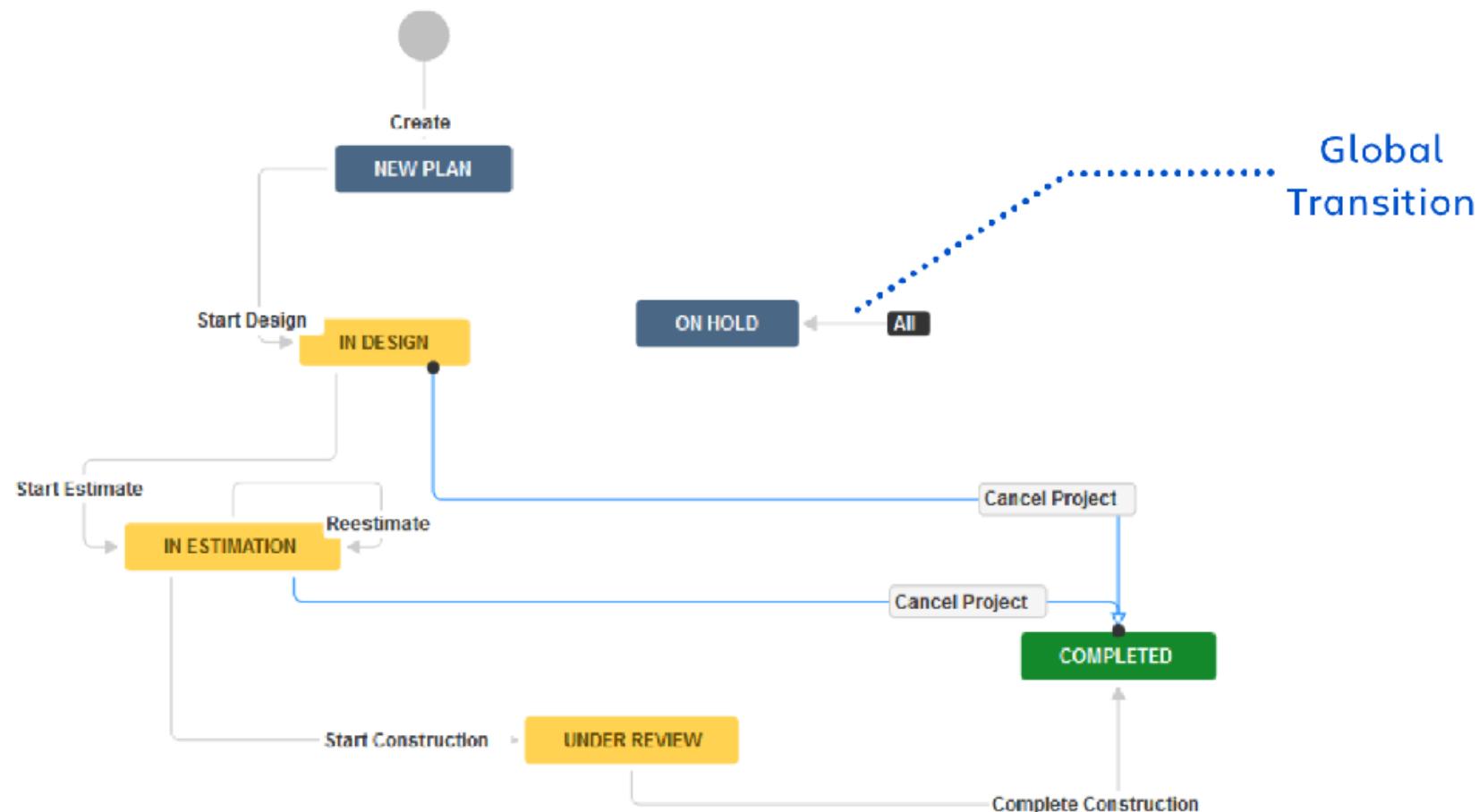
Regular Transitions



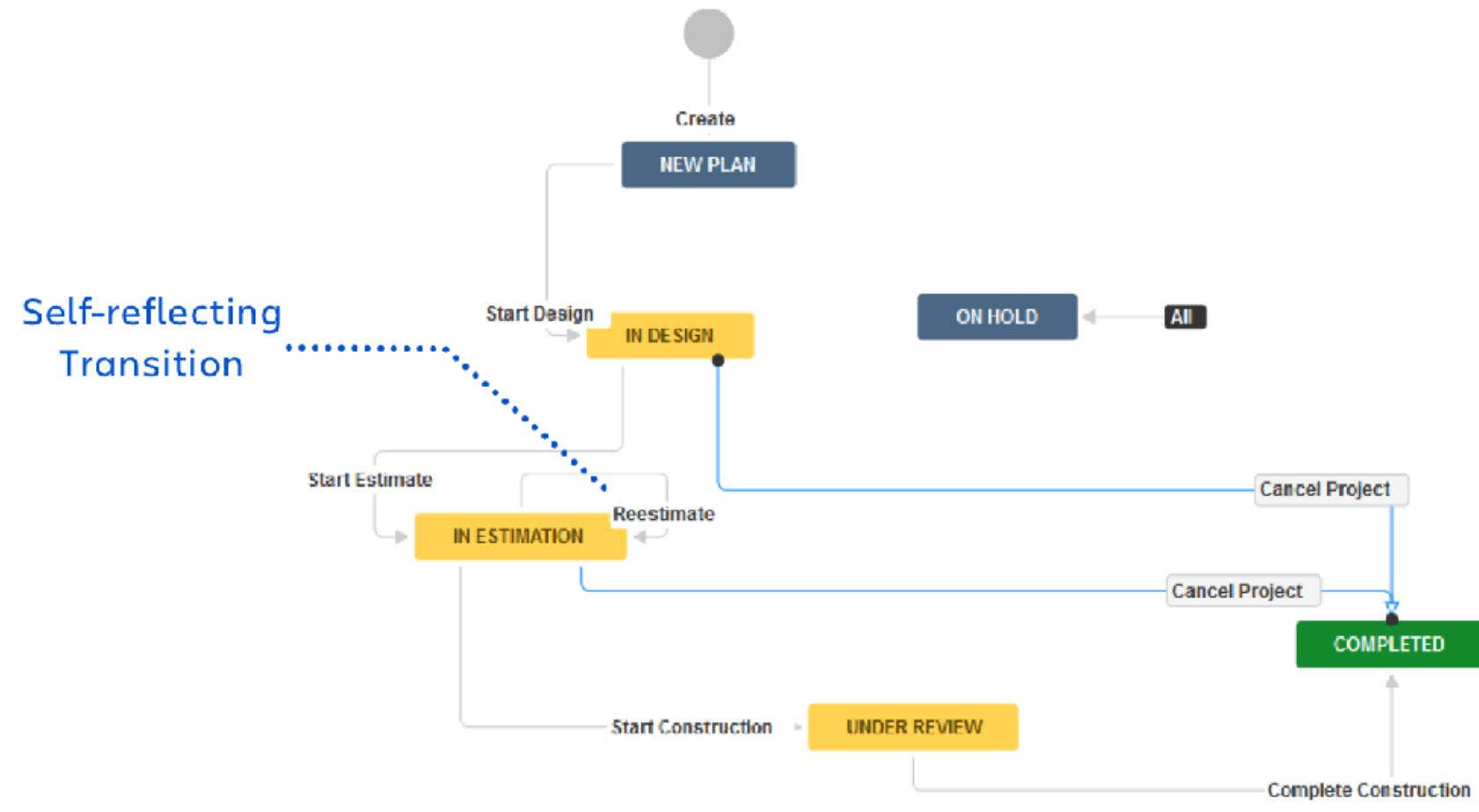
Common Transitions



Global Transitions



Self-reflecting Transitions



Configuring Transitions

Conditions

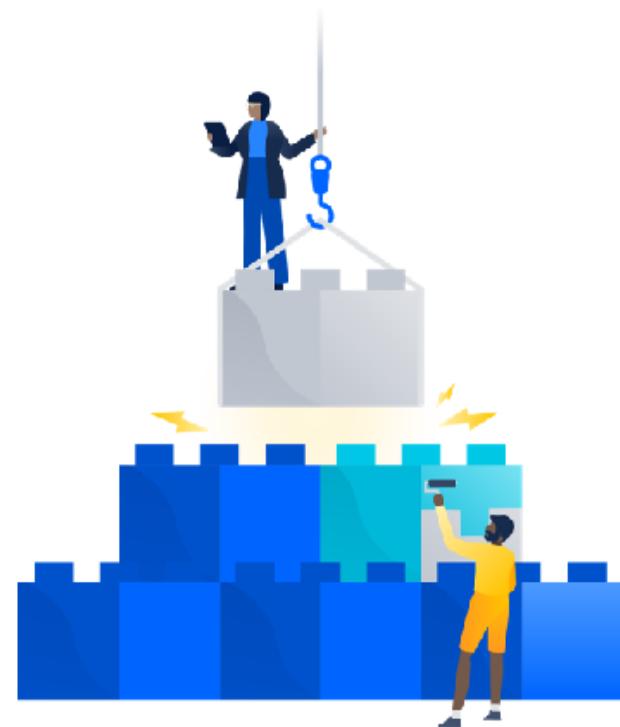
Validators

Post Functions

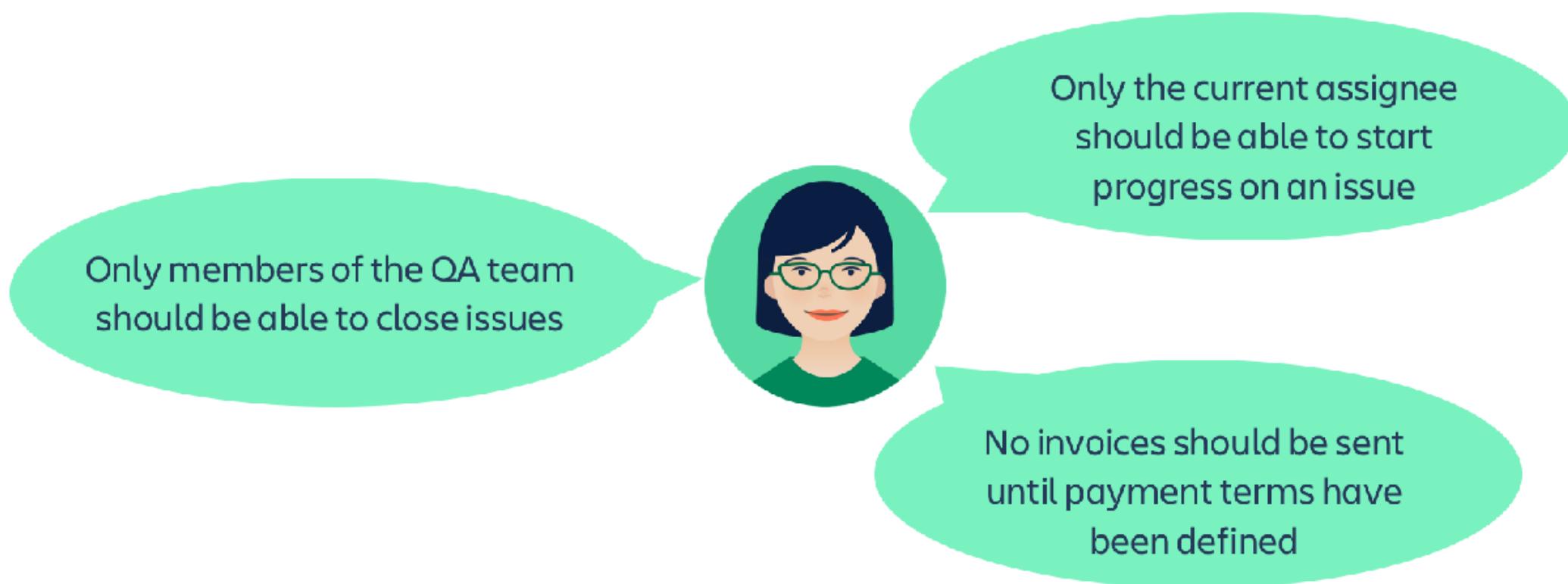
Triggers

Properties

Create Transition



Common Requirements



What do these requirements have in common?



Common Requirements

Only **members of the QA team** should be able to close issues



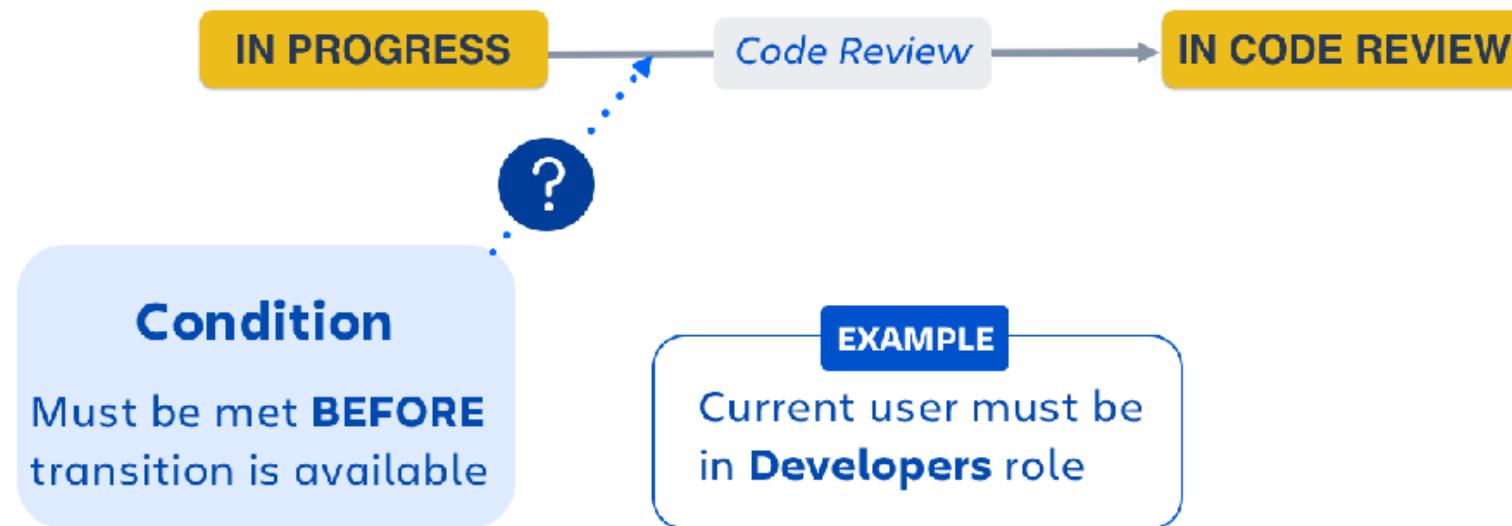
Nobody but the **current assignee** of an issue should be able to start progress on an issue

We can't send out an invoice until **payment terms** have been **defined**

*They're all situations that must exist **before** a transition can be executed*



Conditions Occur Before



If a condition fails, the user will not see that transition button in the issue.



Common Conditions



WHO

- Assignee
- Reporter
- In project role
- In group
- Has permission



WHAT

- Code has been committed
- Status of Sub-tasks



Project Roles vs. Groups

- Membership is for one project
- Project administrators can manage membership, so changes can be made quickly if membership changes
- Helps reduce the number of schemes

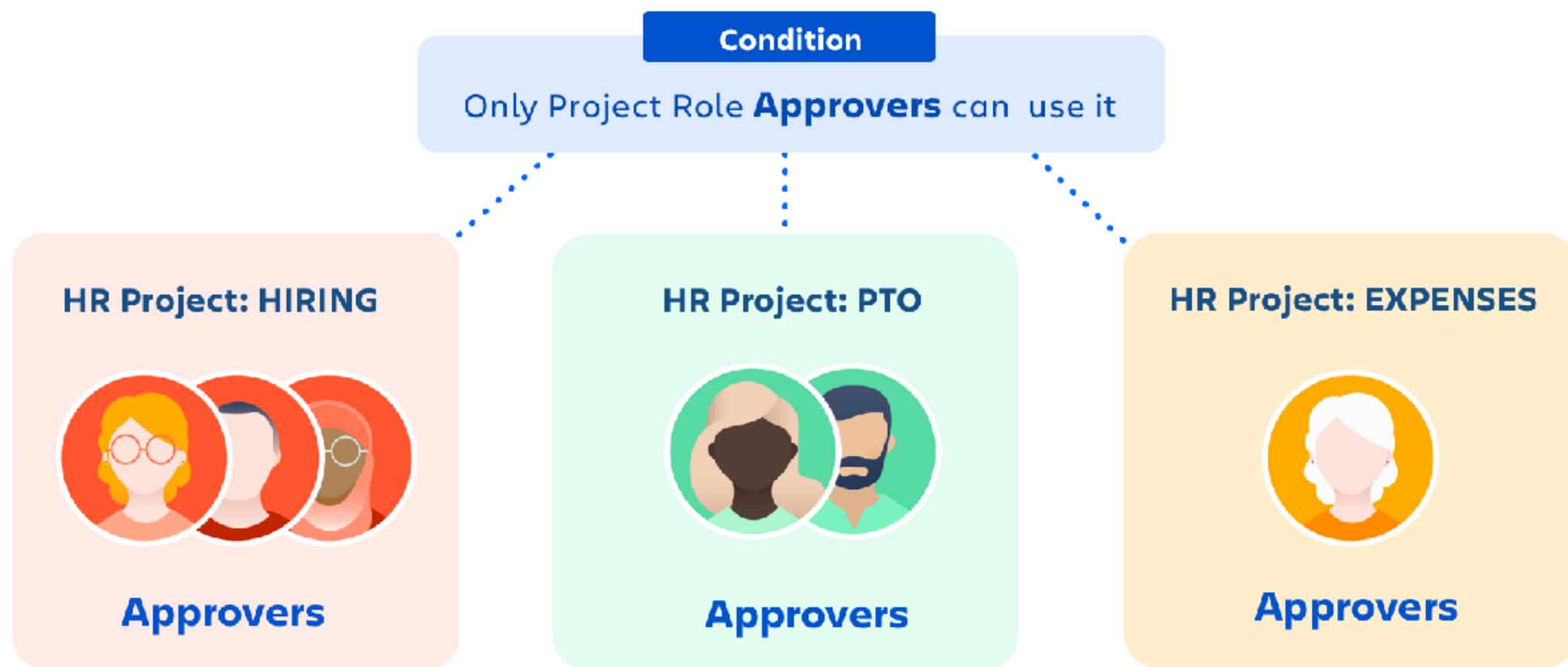
- Membership is global
- Only Jira administrators can manage membership



Only Jira administrators can create project roles and groups



Use Project Roles in Conditions & Validators



Grouping & Nesting Conditions



Who can execute this transition?



Conditions Example

Start Progress transition

1 User is Assignee OR
2 User is in either Developers or Administrators project role

There are 3 conditions

All of the following conditions

Any of the following conditions

User Is In Any Roles (JSU)

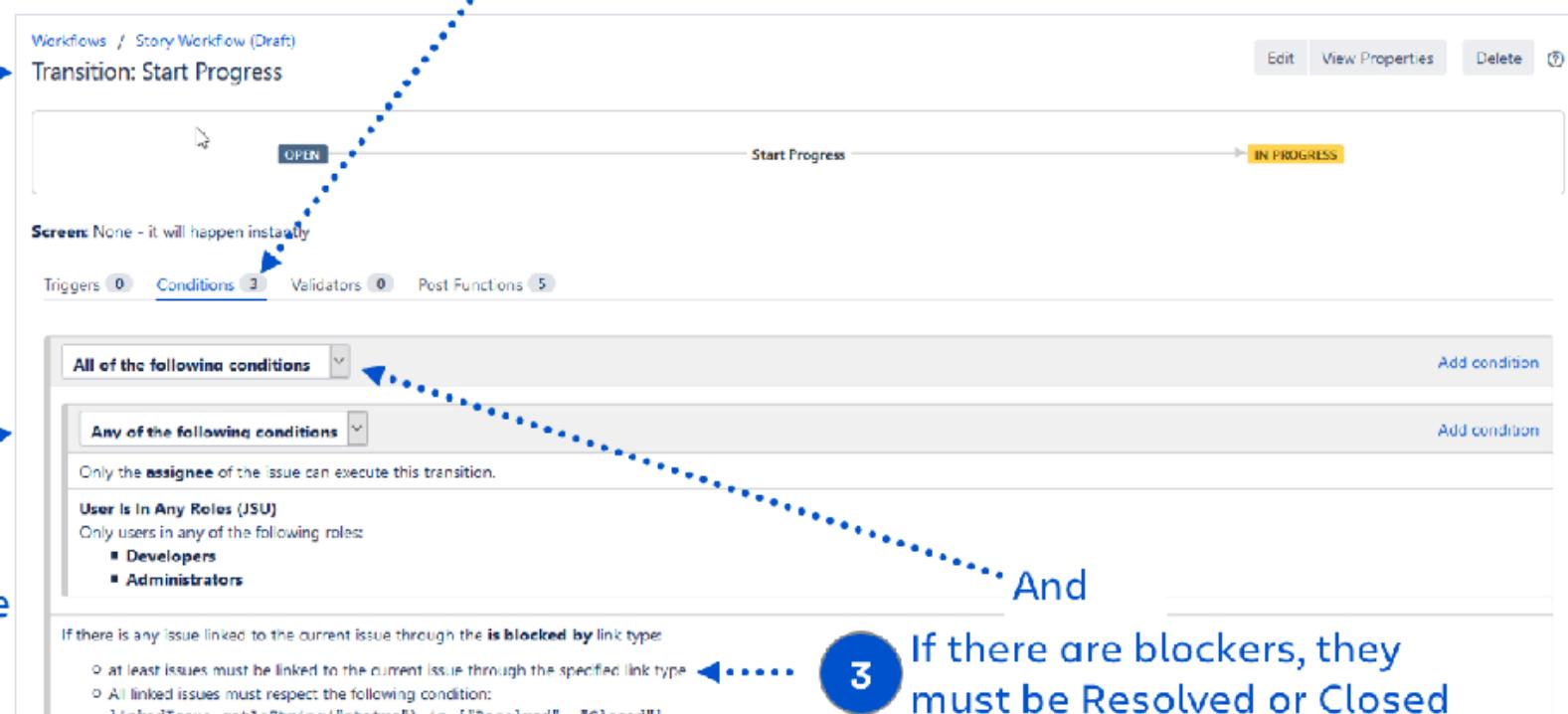
Only users in any of the following roles:

- Developers
- Administrators

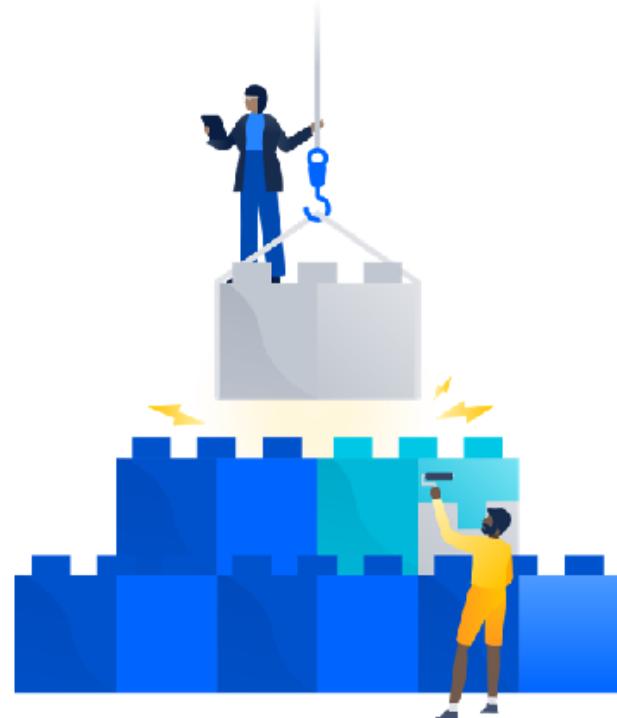
If there is any issue linked to the current issue through the is blocked by link type

- at least issues must be linked to the current issue through the specified link type
- All linked issues must respect the following condition:
`linkedIssue.getString("status") in ["Resolved", "Closed"]`

3 If there are blockers, they must be Resolved or Closed



The screenshot shows the 'Workflows / Story Workflow (Draft)' screen with the 'Transition: Start Progress' selected. The transition starts from the 'OPEN' state and ends at the 'IN PROGRESS' state. A dotted arrow points from the 'Start Progress' transition to the 'Conditions' tab in the navigation bar, which is highlighted in blue. Below the tabs, the condition configuration is shown. It uses a 'All of the following conditions' dropdown. Inside, there is an 'Any of the following conditions' dropdown. Under 'User Is In Any Roles (JSU)', it lists 'Developers' and 'Administrators'. At the bottom, there is a condition for linked issues: 'If there is any issue linked to the current issue through the is blocked by link type'. It specifies that at least one issue must be linked and that all linked issues must have a status of 'Resolved' or 'Closed'. A blue callout box labeled '3' points to this condition with the text 'If there are blockers, they must be Resolved or Closed'.



Configuring Transitions

Conditions

Validators

Post Functions

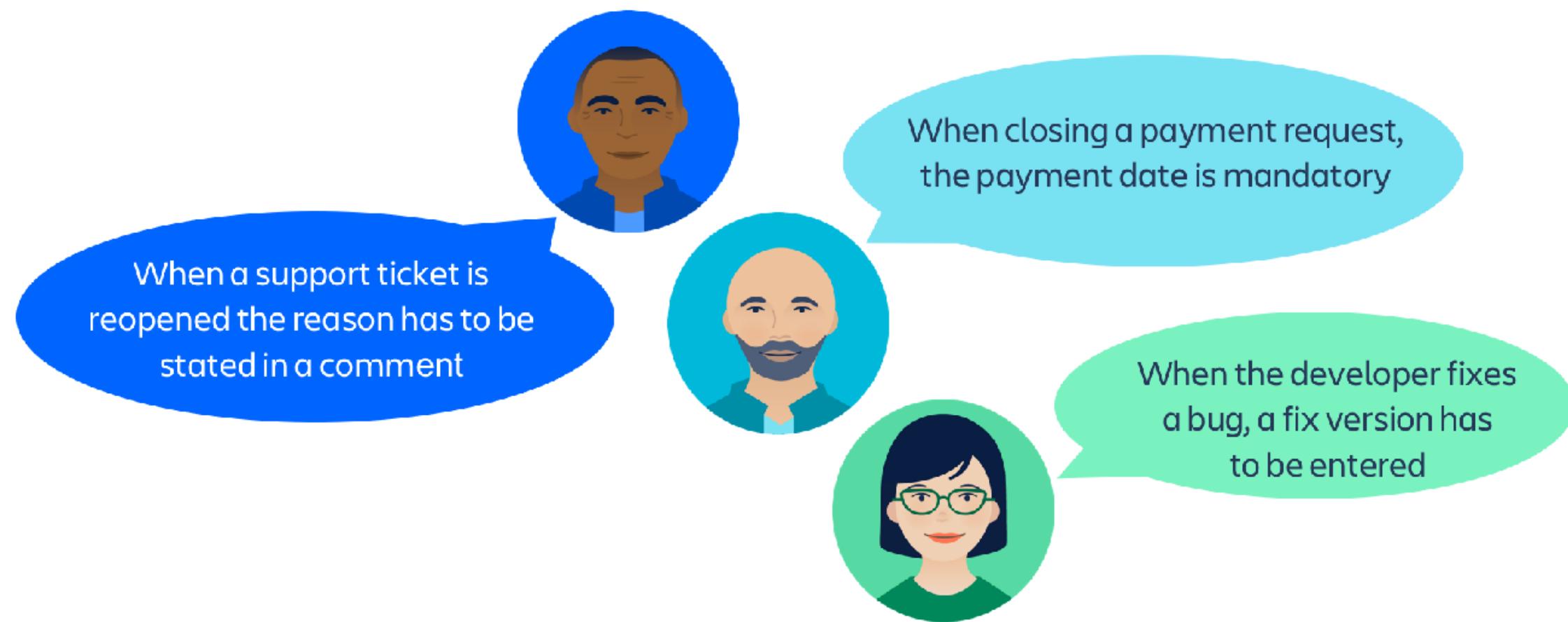
Triggers

Properties

Create Transition



Common Requirements



Common Requirements



When a support ticket is reopened the **reason** has to be stated in a comment



When closing a payment request, the **payment date** is mandatory



When the developer fixes a bug, a **fix version** has to be entered

*Any **input** made to a transition has to be valid **when** it's being executed*



Validators Occur During



- i** *If a validator fails, workflow errors are presented, the issue doesn't progress to the next status, and the post functions aren't executed*



Common Validators

Validator	Examples
Fields Required	Root Cause when resolving an Outage; Comments when rejecting a Request
Regular Expression Check	Email address contains @ symbol
Date Compare	Creation date < Due date



Transition Screens

- Transition screens let users update data as an issue move through a transition
- Jira supplies some transition screens; only Jira administrators can create custom ones



Custom Transition Screen Examples

- Comment field is added automatically if a user has Comment permissions

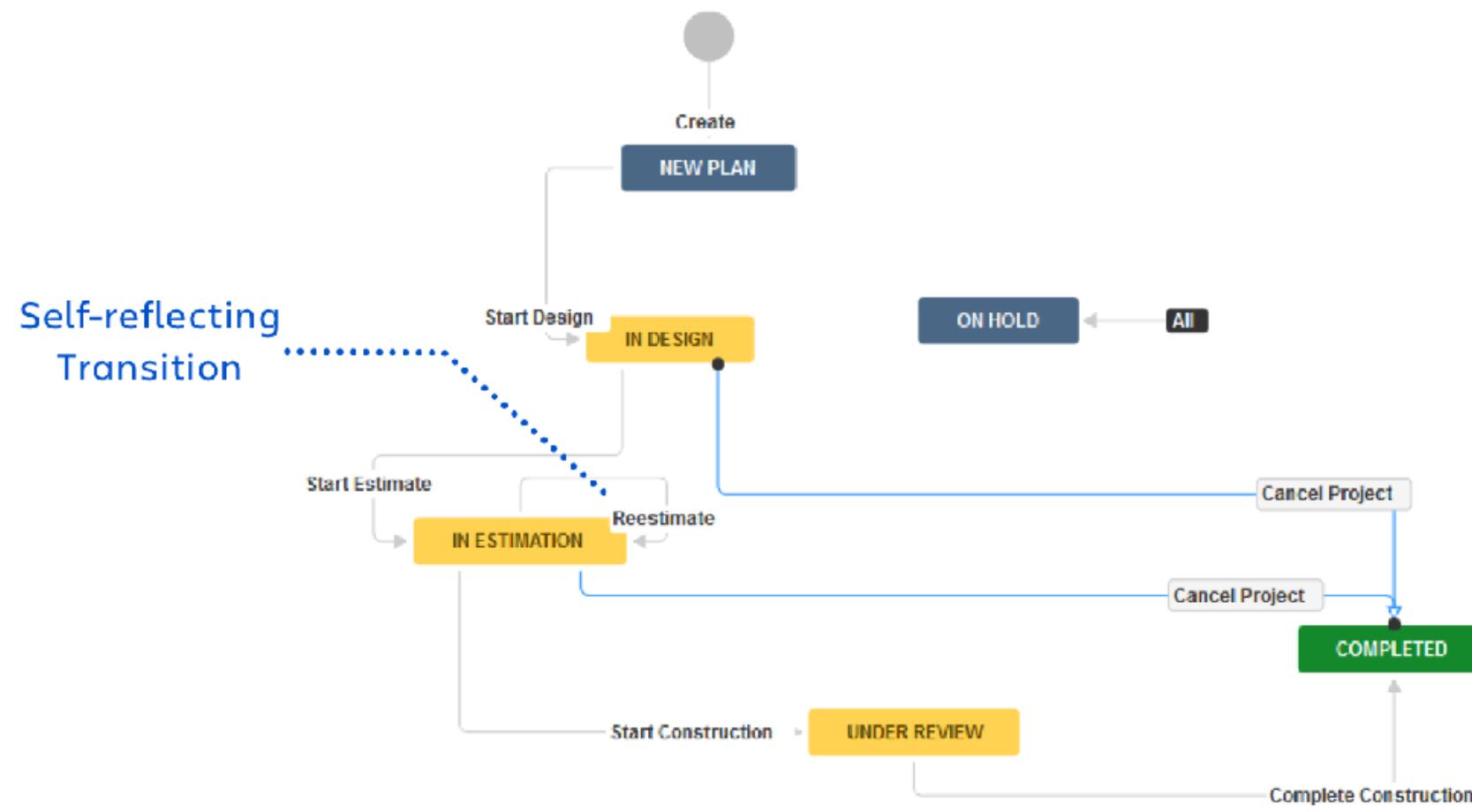
The image displays two Jira custom transition screens side-by-side.

Start Progress: This screen contains a single "Comment" field with rich text editing tools. A blue annotation at the bottom left states: "This screen only contains Comment".

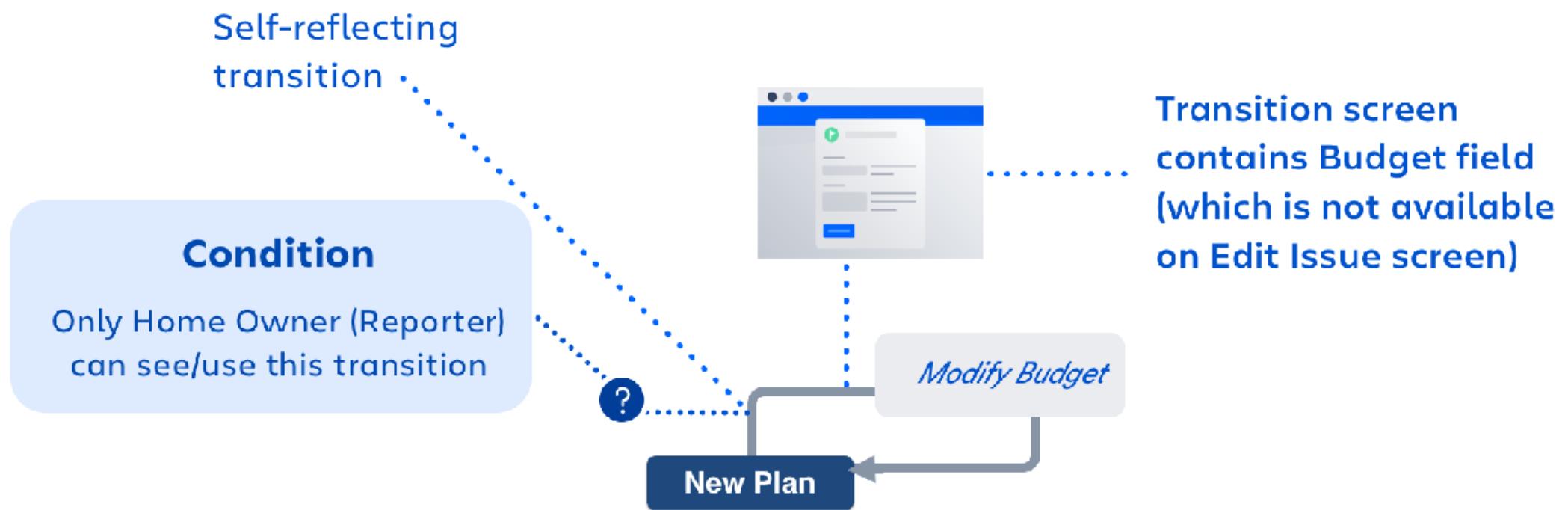
Pay & Close: This screen contains several fields:

- Assignee: Unassigned
- Active: None (radio button selected)
- Account: None
- Pay Date: [empty field]
- Time Spent: [empty field] (e.g. 3w 4d 12h)
- Date Started: 27/Mar/19 9:56 AM
- Remaining Estimate:
 - Adjust automatically (radio button selected)
 - Leave estimate unset
 - Set to: [empty field] (e.g. 3w 4d 12h)
 - Reduce by: [empty field] (e.g. 3w 4d 12h)
- Resolution: Please select...
- Comment: A large rich text editor with standard toolbar buttons.

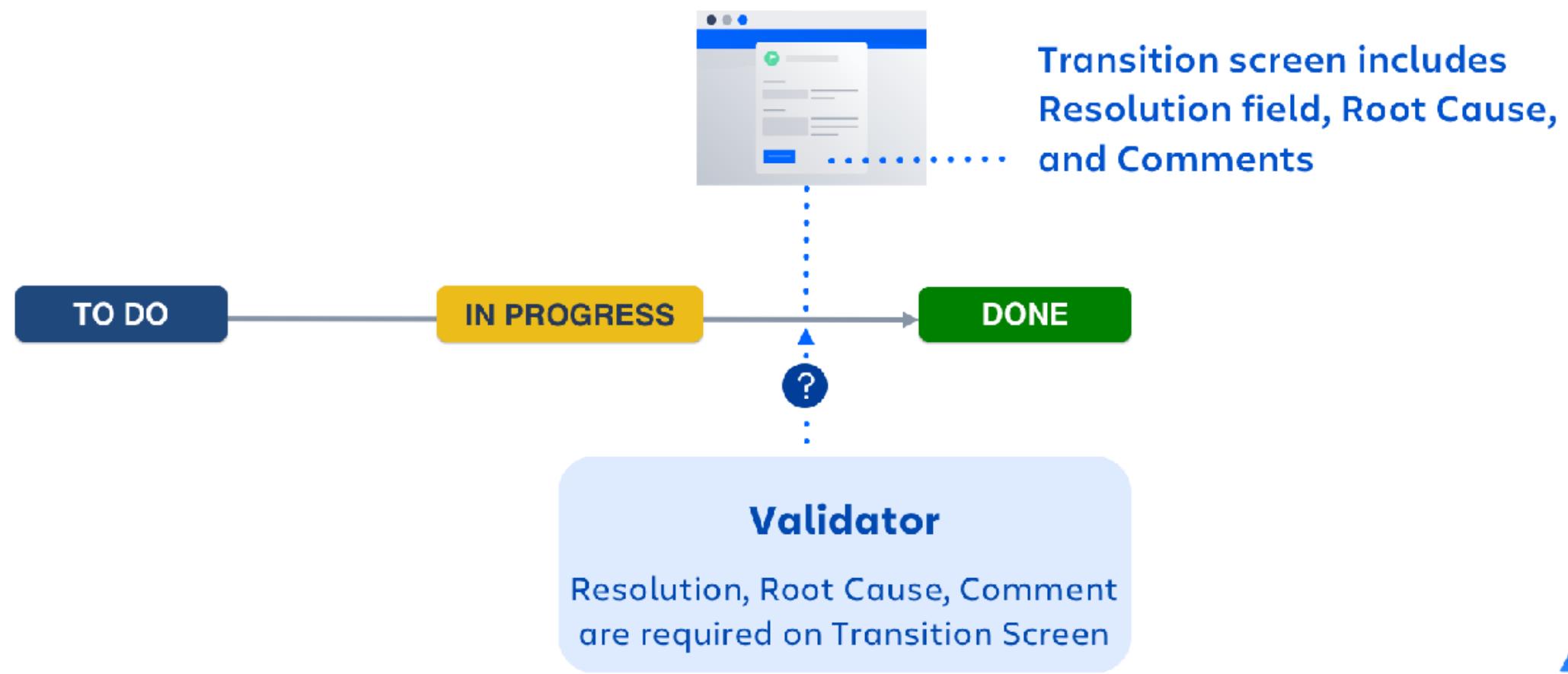
Using Self-reflecting Transitions



Using Conditions to Restrict Fields



Using Validators and Transition Screens



Validators Example

Review transition

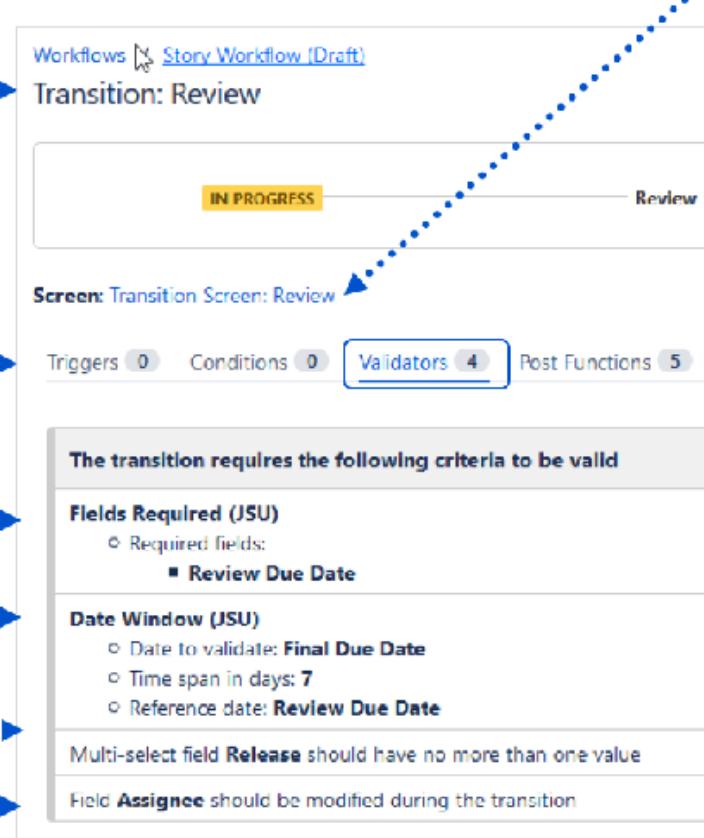
4 validations occur during this transition

1 Review Due Date is required

2 Review Due Date must be within 7 days of Final Due Date

3 Single Release was selected

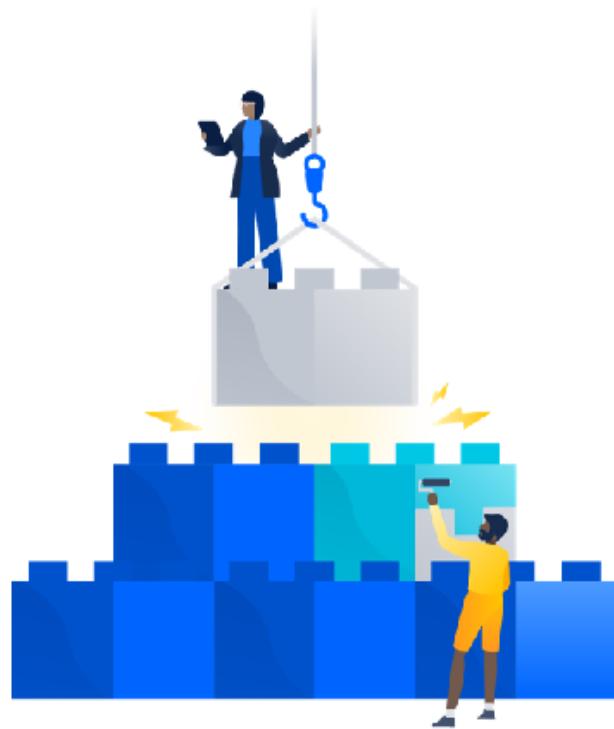
4 Assignee was modified



Transition Screen

The screenshot shows the "Review" transition screen. It includes fields for Assignee (Anne), Release (G1 Release), Review Due Date (21/Mar/19), and Final Due Date (29/Mar/19). The comment field contains the text: "Please focus on the discrepancies in draft.".





Configuring Transitions

Conditions

Validators

Post Functions

Triggers

Properties

Create Transition



Common Requirements

Bugs should be automatically assigned to a QA Engineer as soon as they're fixed



For reports, the current date needs to be captured automatically once a product has shipped

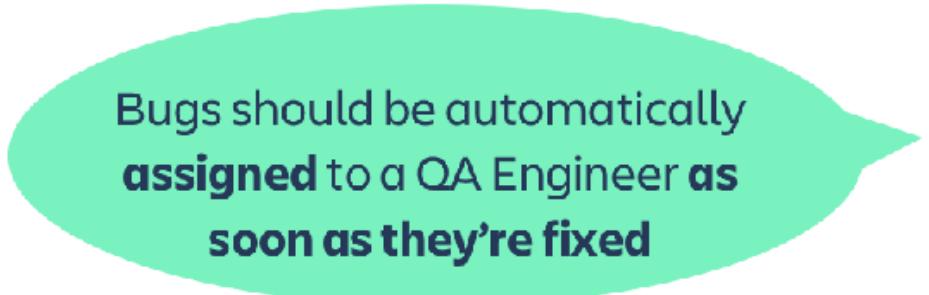
When product support reopens a request, the resolution should be cleared automatically



What do these requirements have in common?



Common Requirements



Bugs should be automatically assigned to a QA Engineer as soon as they're fixed



For reports, the current date needs to be captured automatically once a product has shipped

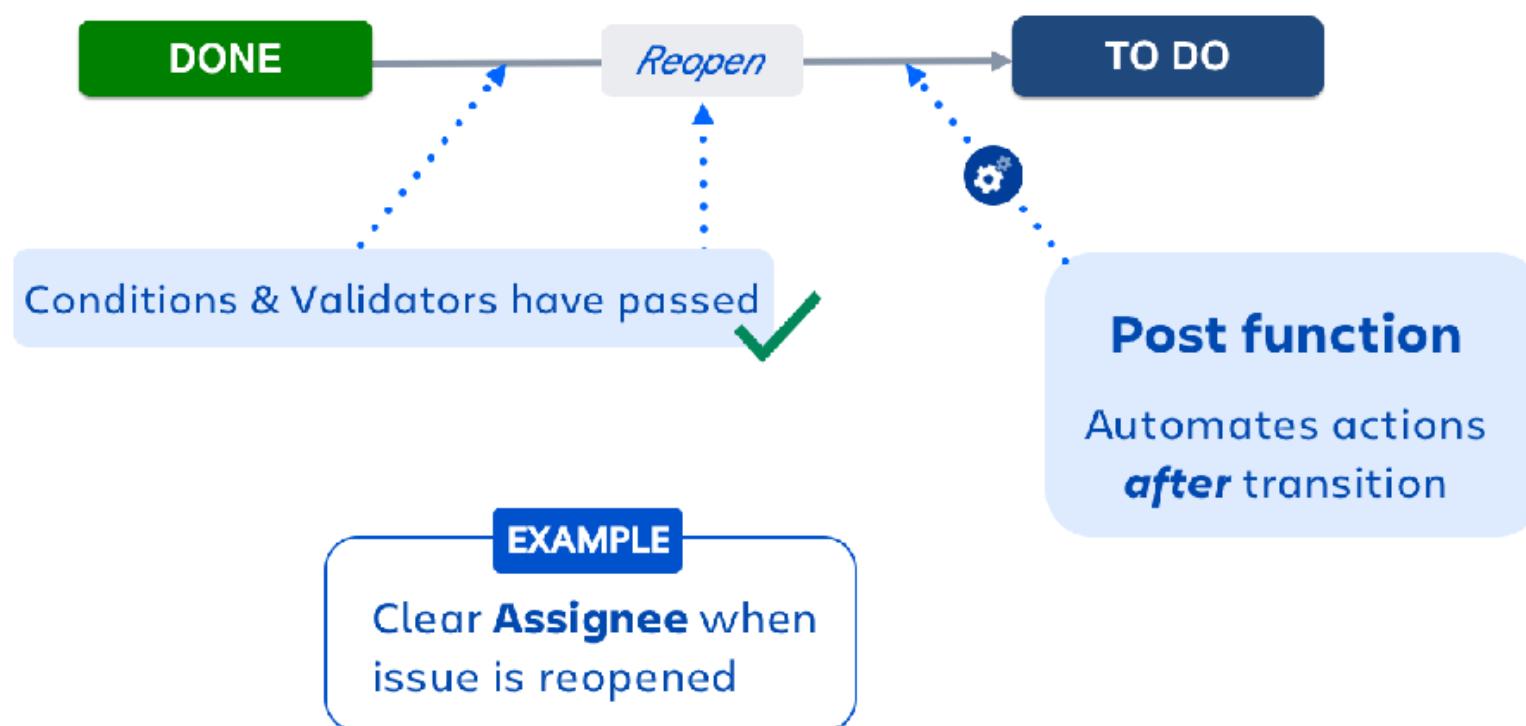


When product support reopens a request, the resolution should be cleared automatically

*These are **automated actions** performed **after** something has occurred*



Post Functions Occur After



Common Post Functions

Assign issues



Reporter

Update fields

Priority: **Highest**

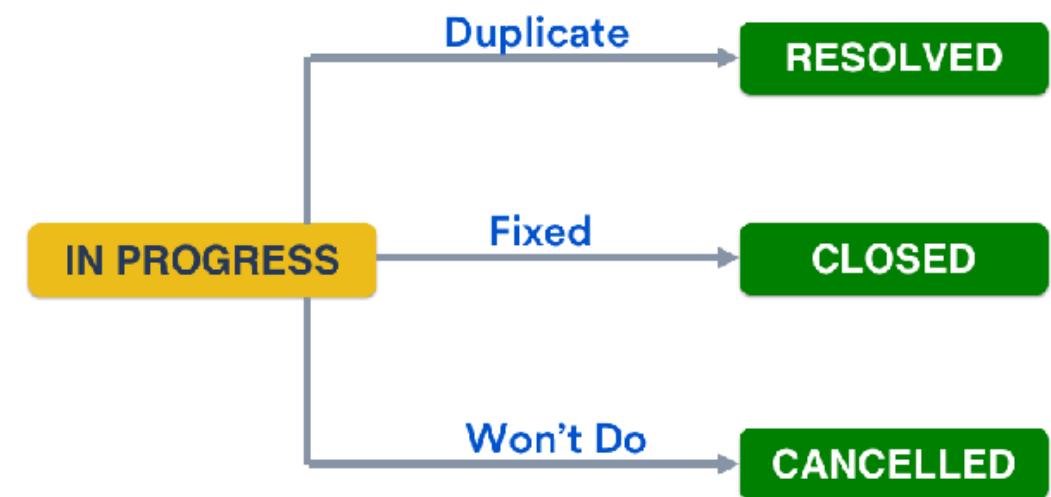


Setting the Resolution Field

User selects Resolution via Transition Screen



Resolution set automatically via Post Function

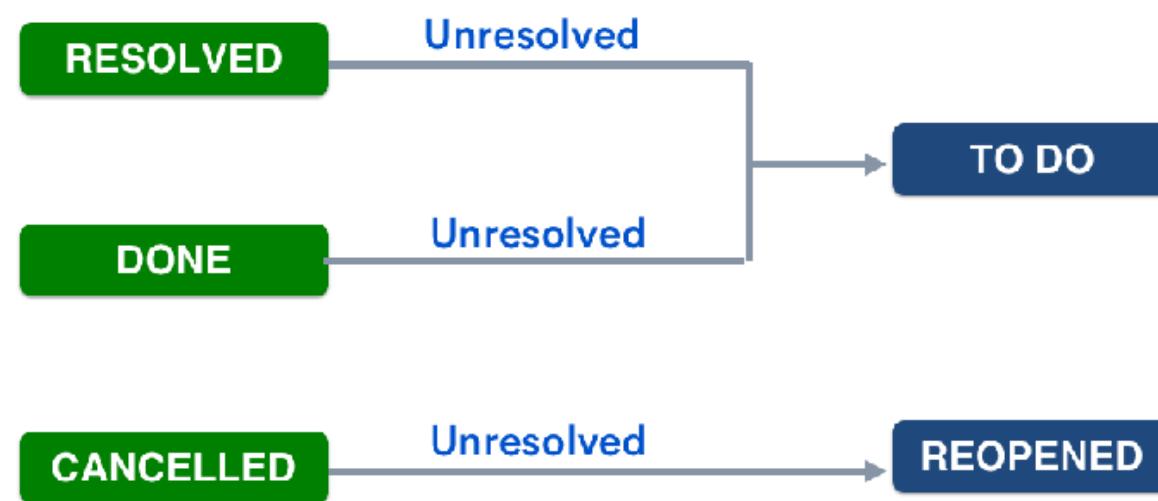


Ensure resolution is set by either a post function or in a transition screen.



Clearing the Resolution Field

Resolution cleared automatically via Post Function



Ensure resolution is cleared whenever issue is being reopened.



5 Essential Post Functions

The following will be processed after the transition occurs

1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. Re-index an issue to keep indexes in sync with the database.
5. Fire a **Generic Event** that can be processed by the listeners.

..... Generic Event is fired by
default on new transitions



Can't be deleted or reordered. Optional post functions can be inserted before/between.



Ordering Post Functions

Field changes should precede update of change history

The following will be processed after the transition occurs

1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. Update change history for an issue and store the issue in the database.
4. The **Resolution** of the issue will be set to **Duplicate**.
5. Re-index an issue to keep indexes in sync with the database.
6. Fire a **Generic Event** that can be processed by the listeners.



Correct Order of Post Functions

Order of
Post
Functions
is fixed

The following will be processed after the transition occurs

1. Set issue status to the linked status of the destination workflow step.
2. Add a comment to an issue if one is entered during a transition.
3. The **Resolution** of the issue will be set to **Duplicate**.
4. Update change history for an issue and store the issue in the database.
5. Re-index an issue to keep indexes in sync with the database.
6. Fire a **Generic Event** that can be processed by the listeners.



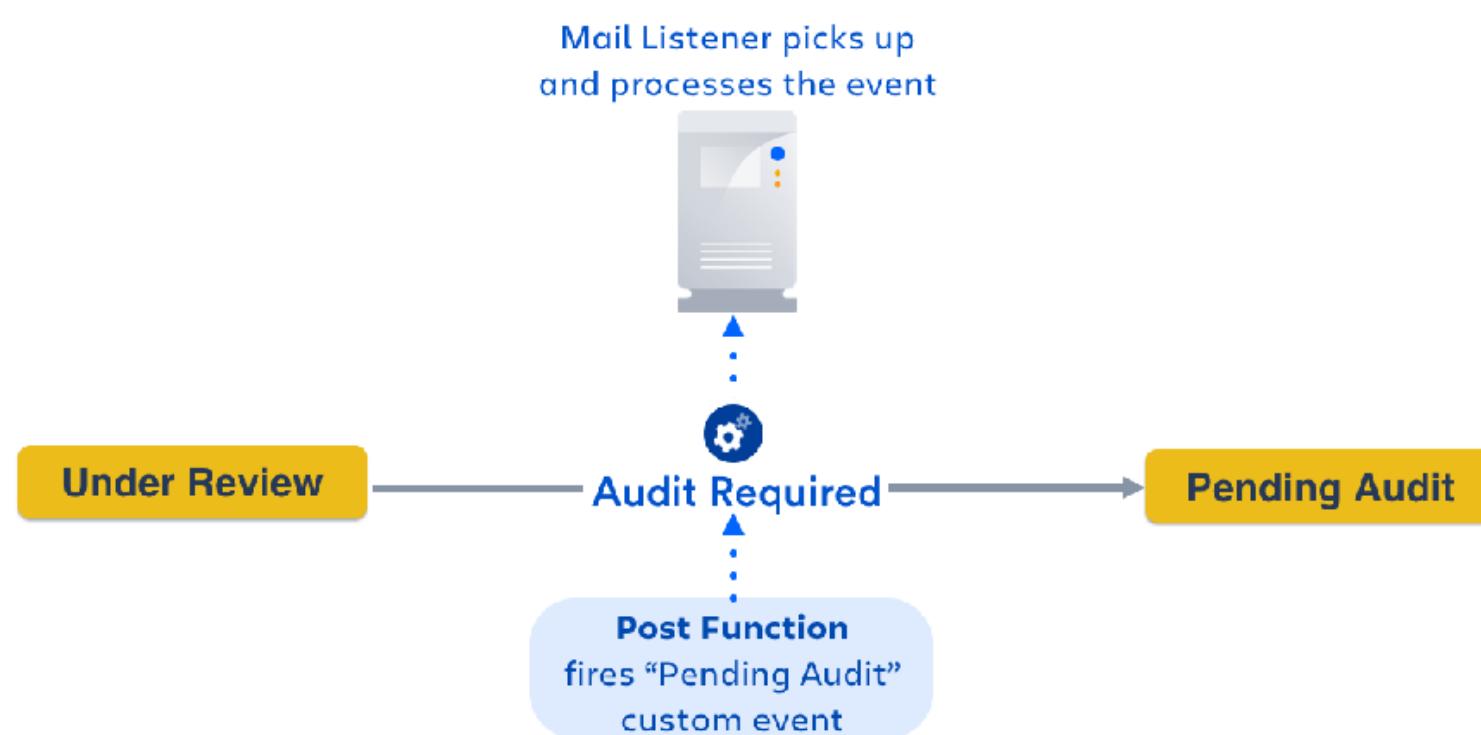
The sequence of post functions matters.



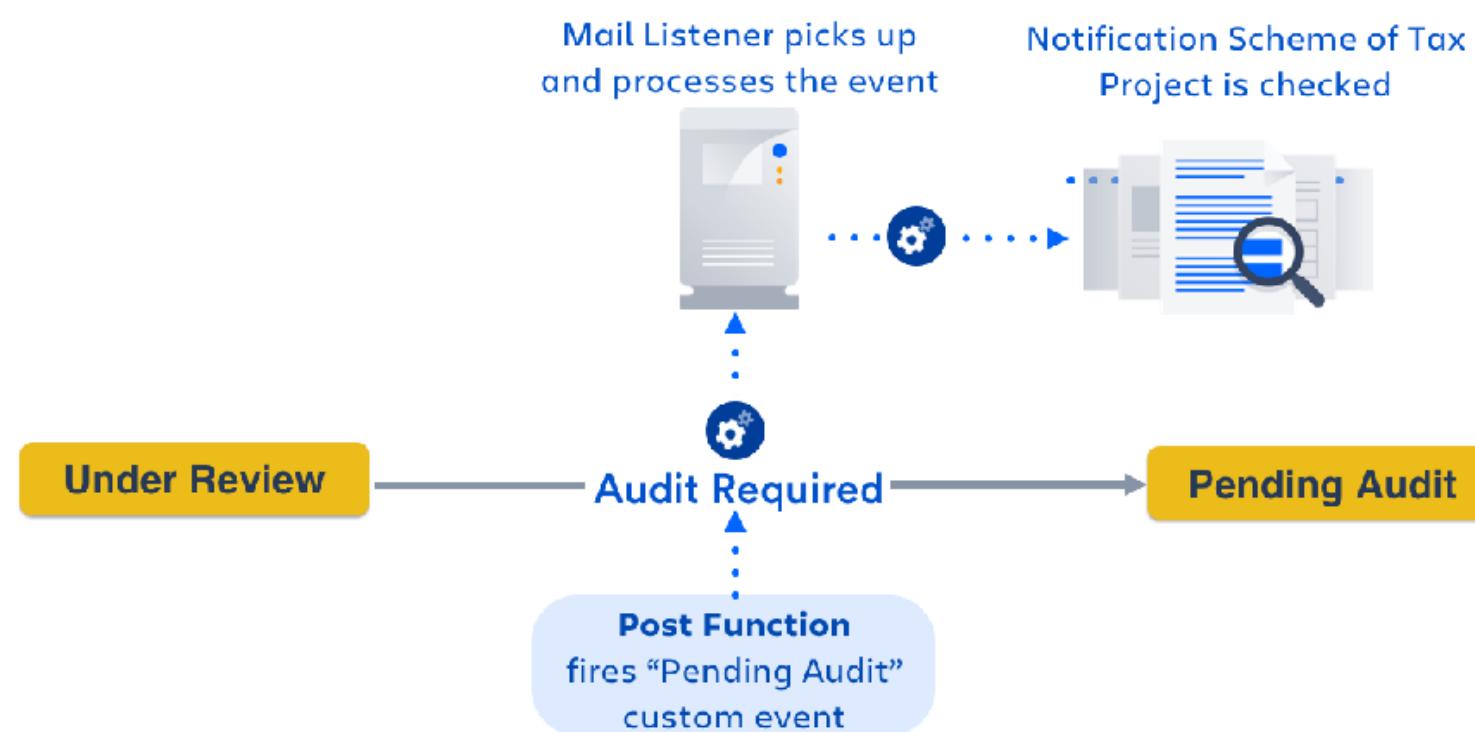
Using Custom Events



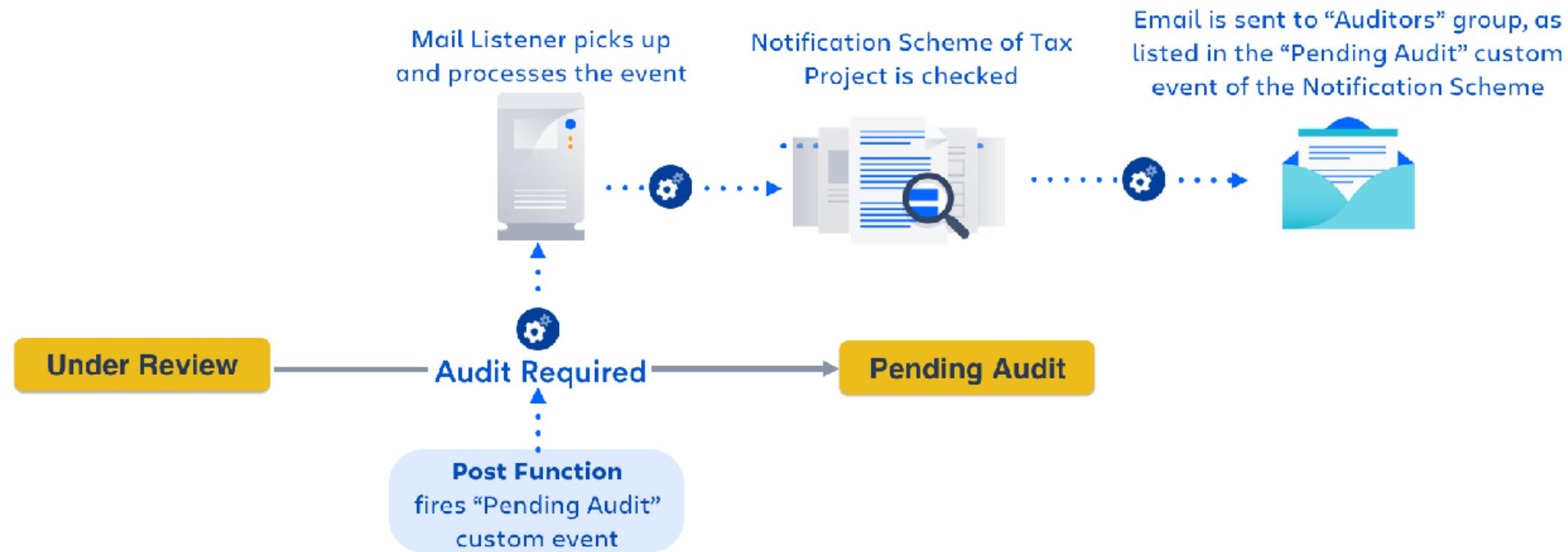
Using Custom Events



Using Custom Events



Using Custom Events



Using Custom Events



To use a Custom Event, you must select it in a transition Post Function and add entries to that event in the appropriate Notification Scheme.



SYSTEM EVENTS

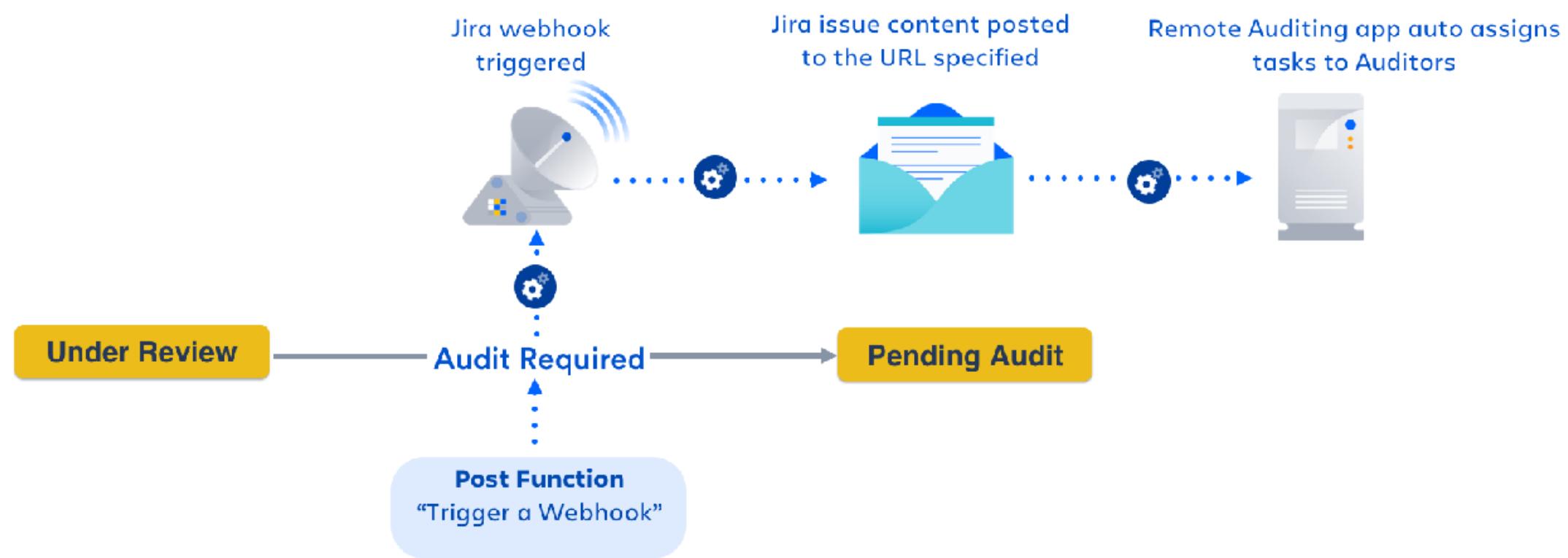
Issue Created	Issue Updated	Issue Assigned
Issue Resolved	Issue Closed	Issue Commented
Issue Comment Edited	Issue Comment Deleted	Work Logged On Issue
Issue Worklog Updated	Issue Worklog Deleted	Work Started On Issue
Work Stopped On Issue	Issue Deleted	Issue Moved
	Issue Reopened	Generic Event

CUSTOM EVENTS

Awaiting Response	Pending Audit	Issue Rejected
Customer Notified		



Using WebHooks



Unselect all events from the webhook admin screen to prevent triggering twice.



Post Functions Example

Transition: Audit Required



Screen: [Transition Screen: Review](#)

7 post functions in transition;
2 added to the 5 default ones

Triggers 0 Conditions 0 Validators 4 Post Functions 7

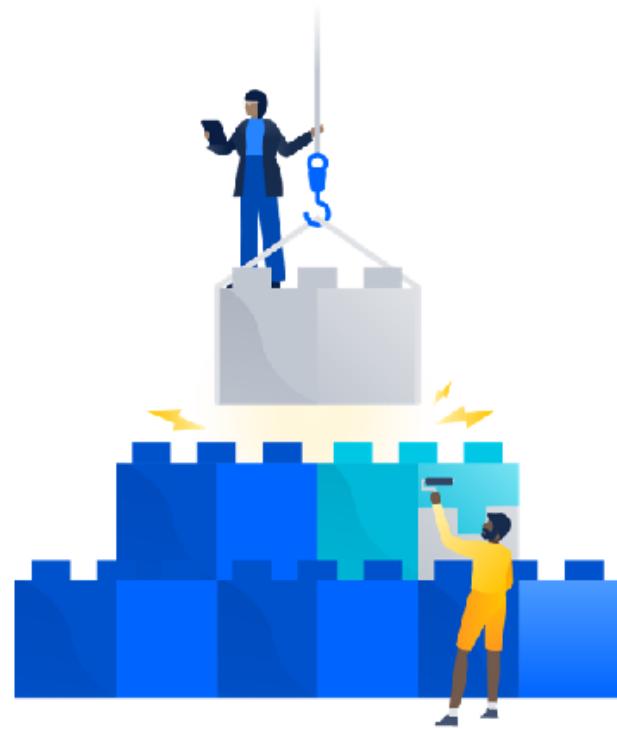
The following will be processed after the transition occurs		Add post function
1. Copy Value From Other Field (JSU) <input type="radio"/> Copy Field Lead Auditor to: Assignee	2. Send transition to webhook: Remote Auditing Application	3. Set issue status to the linked status of the destination workflow step.
4. Add a comment to an issue if one is entered during a transition.	5. Update change history for an issue and store the issue in the database.	6. Re-index an issue to keep indexes in sync with the database.
7. Fire a Pending Audit event that can be processed by the listeners.		

Assign to the Lead Auditor
.....

Trigger webhook
.....

Fire custom event
.....





Configuring Transitions

Conditions

Validators

Post Functions

Triggers

Properties

Create Transition



Common Requirements

Automatically transition Bug to In Progress when a branch is created in Bitbucket.



Automatically transition issue when pull request is created.

Automatically close the task when the review is closed in Crucible.



What do these requirements have in common?



Common Requirements

Automatically transition Bug to In Progress when a branch is created in Bitbucket.



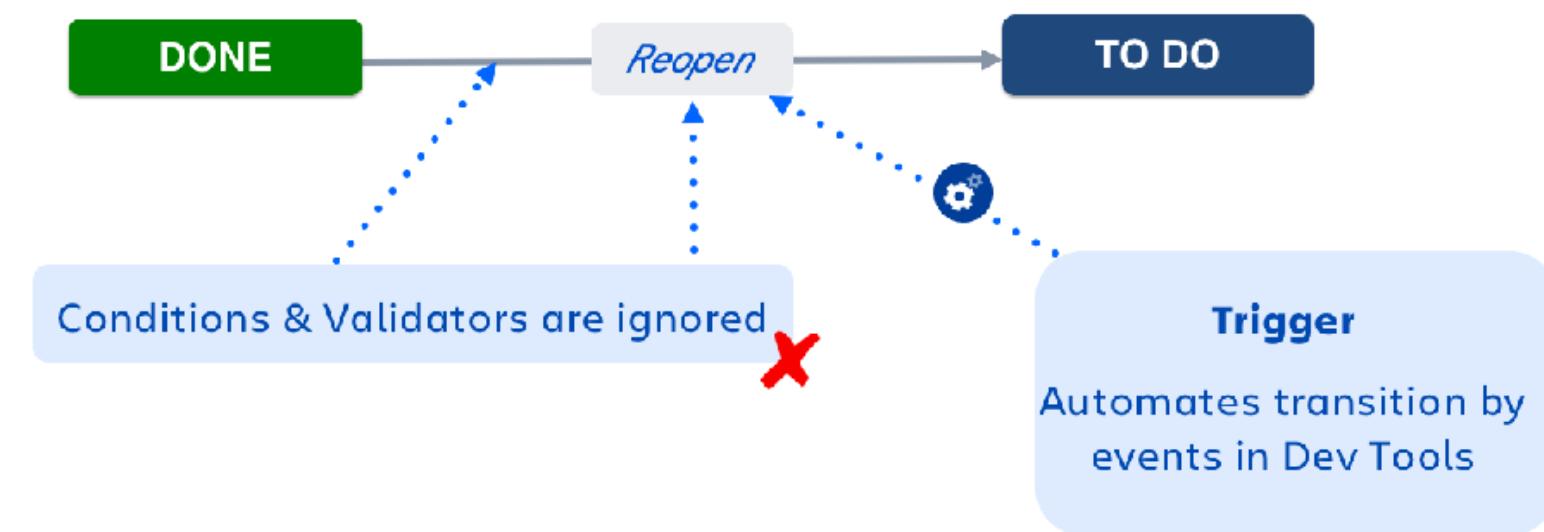
Automatically transition issue when pull request is created.

Automatically close the task when the review is closed in Crucible.

*These are **automated actions** performed in Jira as a result of events in Dev Tools.*



Triggers Automate Transitions



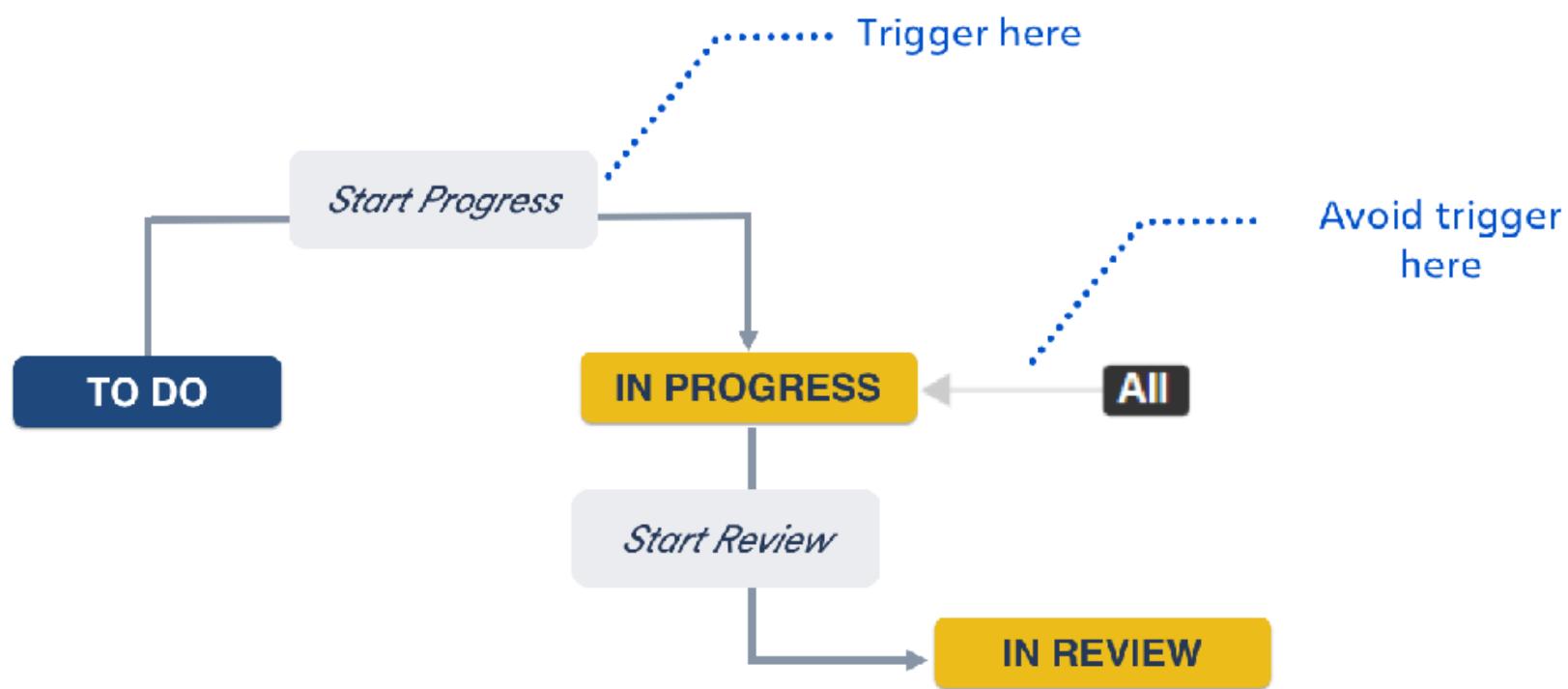
Common Triggers

Dev Tool	Event	Transition Issue
 Bitbucket	• Branch created	TO DO → IN PROGRESS
	• Commit created	IN PROGRESS → IN REVIEW
	• Pull request created	IN REVIEW → DONE
	• Pull request merged	

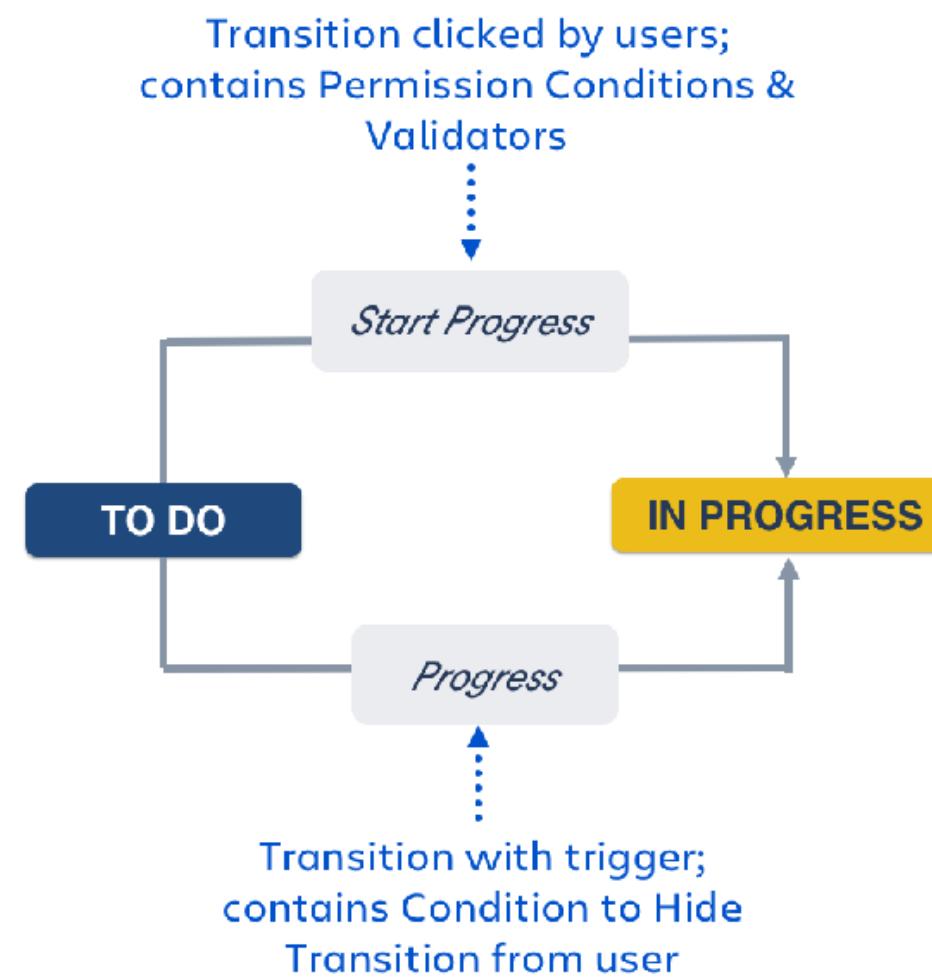
 Crucible	• Review started	IN PROGRESS → IN REVIEW
	• Review closed	IN REVIEW → DONE



Avoid Triggers for Global Transitions



Trigger Tip



Triggers Example

Trigger configured.
No conditions or validators



Branch created in Bitbucket



Transition: Auto Start Progress

Screen: None - it will happen instantly

Triggers 1 Conditions 0 Validators 0 Post Functions 5

A trigger is an event that initiates an automatic transition for a Jira issue. Examples of triggers include the creation of a pull request, the rejection of a code review, and more. [Send feedback on triggers](#).

Caveats: Conditions, validators, and permissions will be ignored for automatic transitions. Global transitions can be automated with triggers, but please [read our guide first](#).

Not sure where to start? [Read our guide on configuring triggers](#).

Add trigger

Trigger	Details	Actions
Branch created	Automatically transitions the issue when a related branch is created in a connected repository	View details



Configuring Transitions

Conditions

Validators

Post Functions

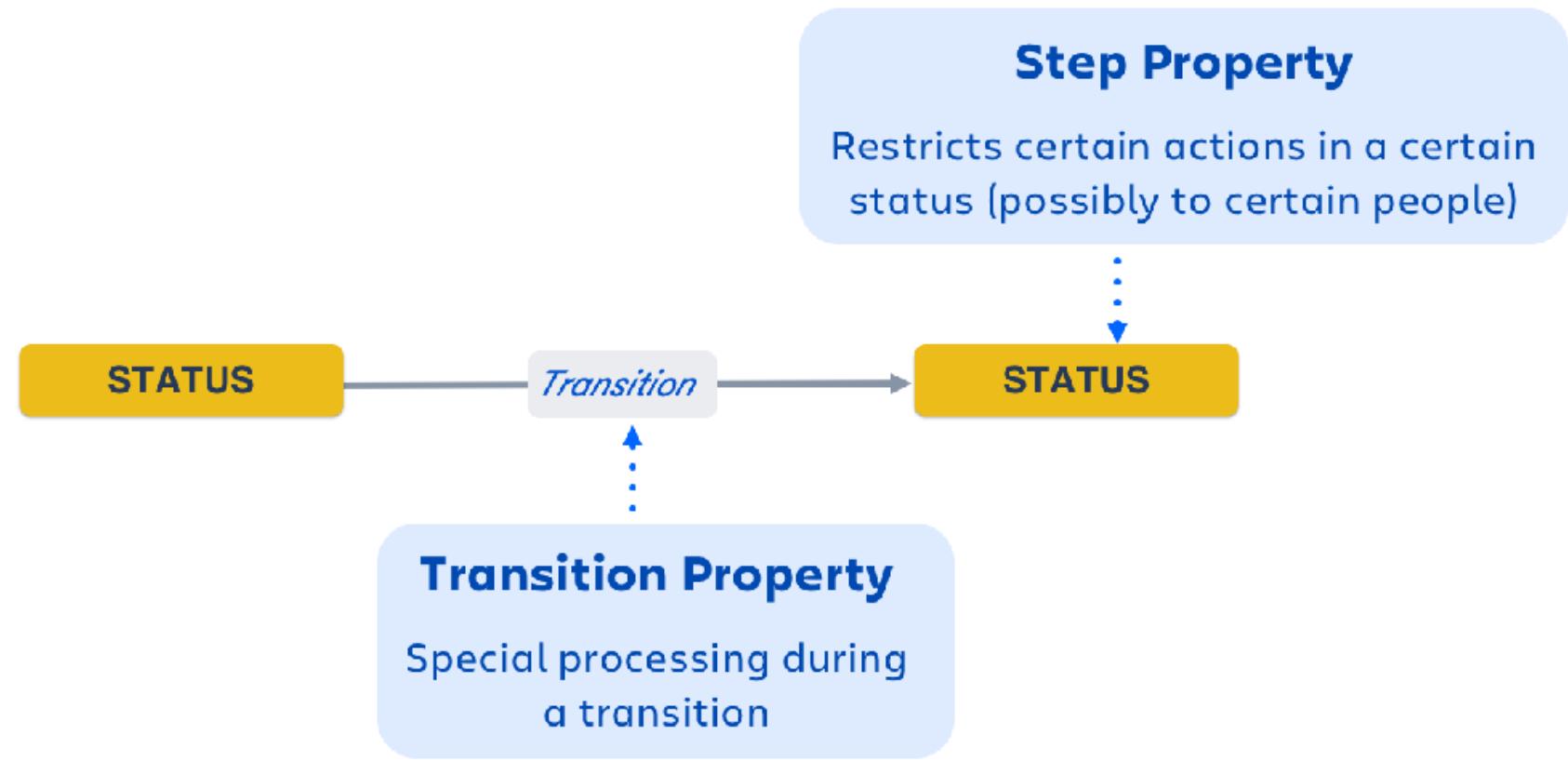
Triggers

Properties

Create Transition



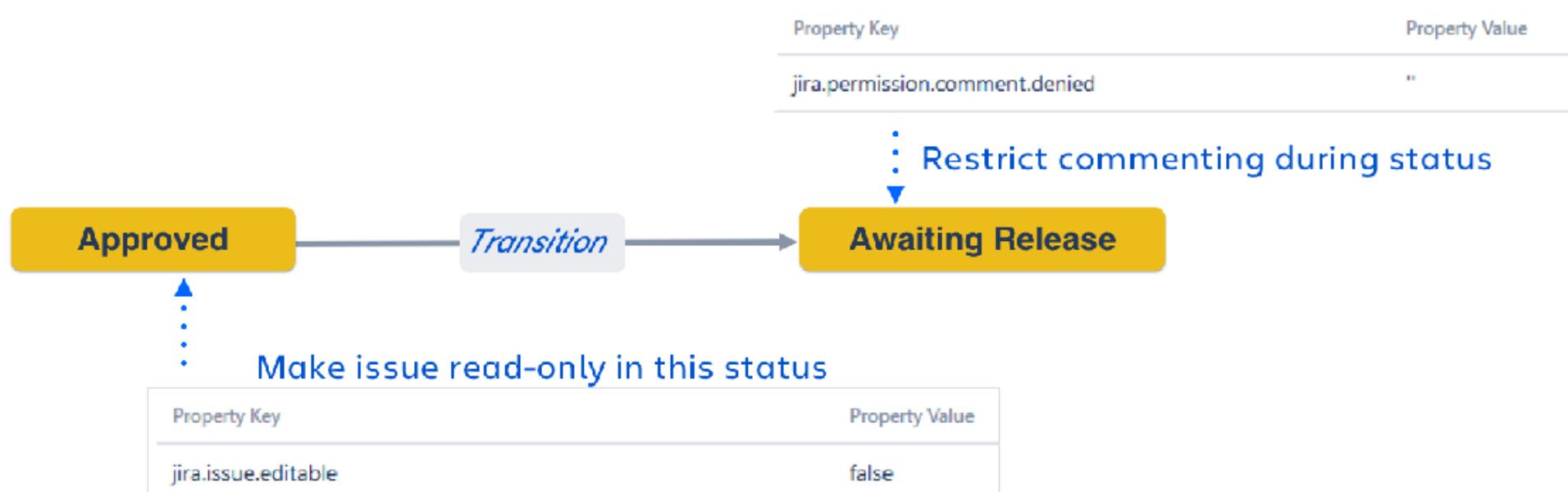
Workflow Properties



Use properties sparingly, if business requirements warrant them.



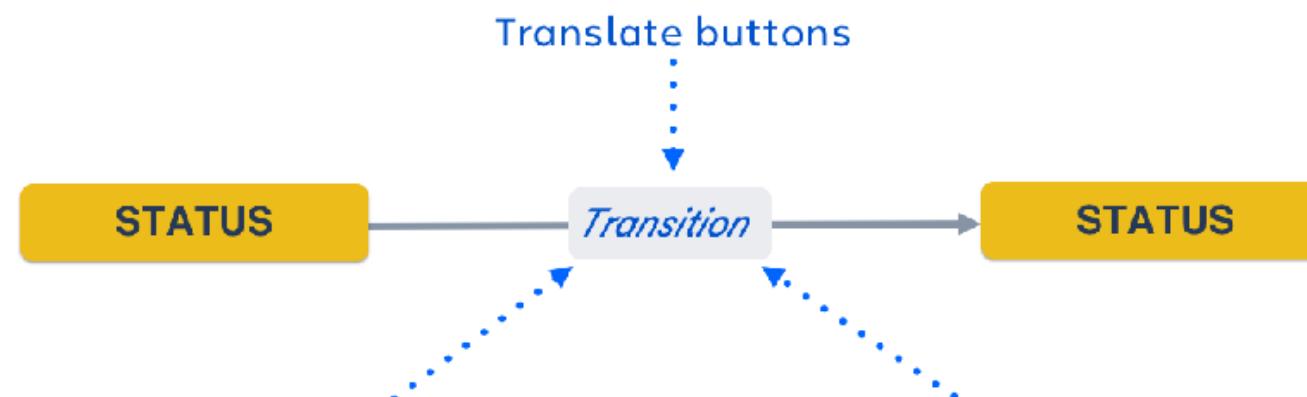
Step Property Examples



Permission workflow property allows more granular control than permission schemes.



Transition Property Examples



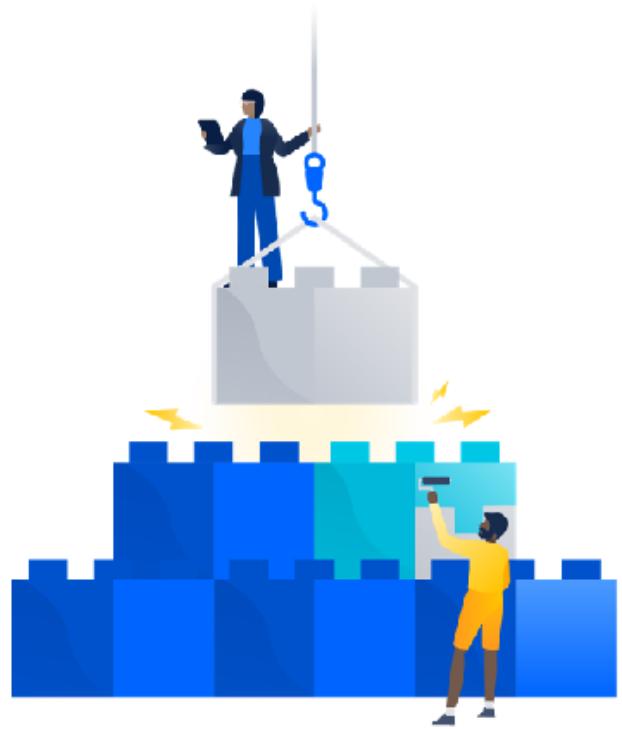
Change button order

Property Key	Property Value
<code>opsbar-sequence</code>	10

Exclude two of the available Resolutions

Property Key	Property Value
<code>jira.field.resolution.exclude</code>	10000,10001





Configuring Transitions

Conditions

Validators

Post Functions

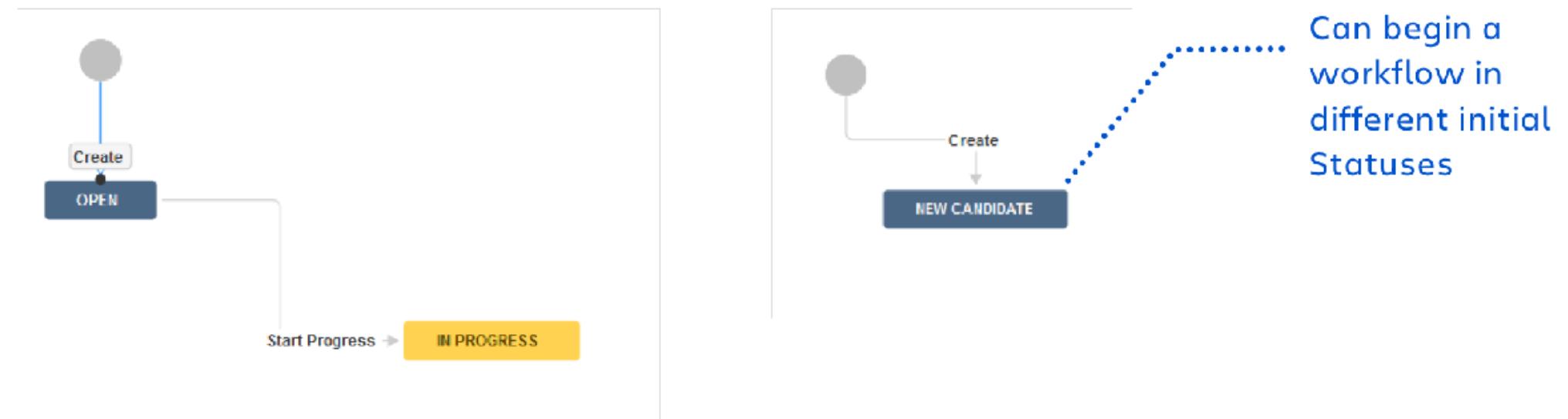
Triggers

Properties

Create Transition



Create Transition Is Automatically Created



The initial transition cannot be deleted.



Create Transition Has No Conditions

Workflows / Test1
Transition: Create

Edit View Properties ⓘ

```
graph LR; Start(( )) -- Create --> Open([OPEN]);
```

This is the **Initial** transition in the workflow.

Screen: None - initial transition does not have a view.

[Validators 1](#) [Post Functions 3](#)

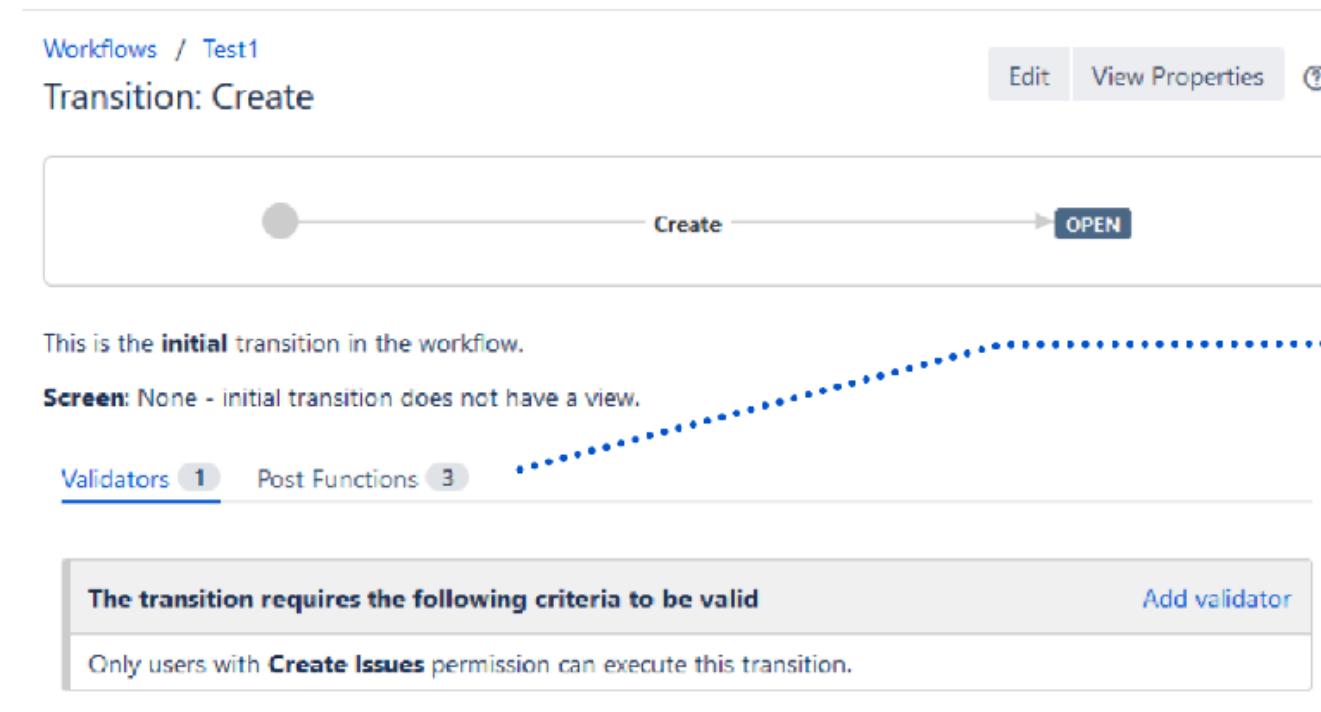
The transition requires the following criteria to be valid

Add validator

Only users with **Create Issues** permission can execute this transition.



Avoid Webhooks in Create Transition



Add post functions but
not webhooks.
Configure them to fire
from the issue_created
event instead



Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

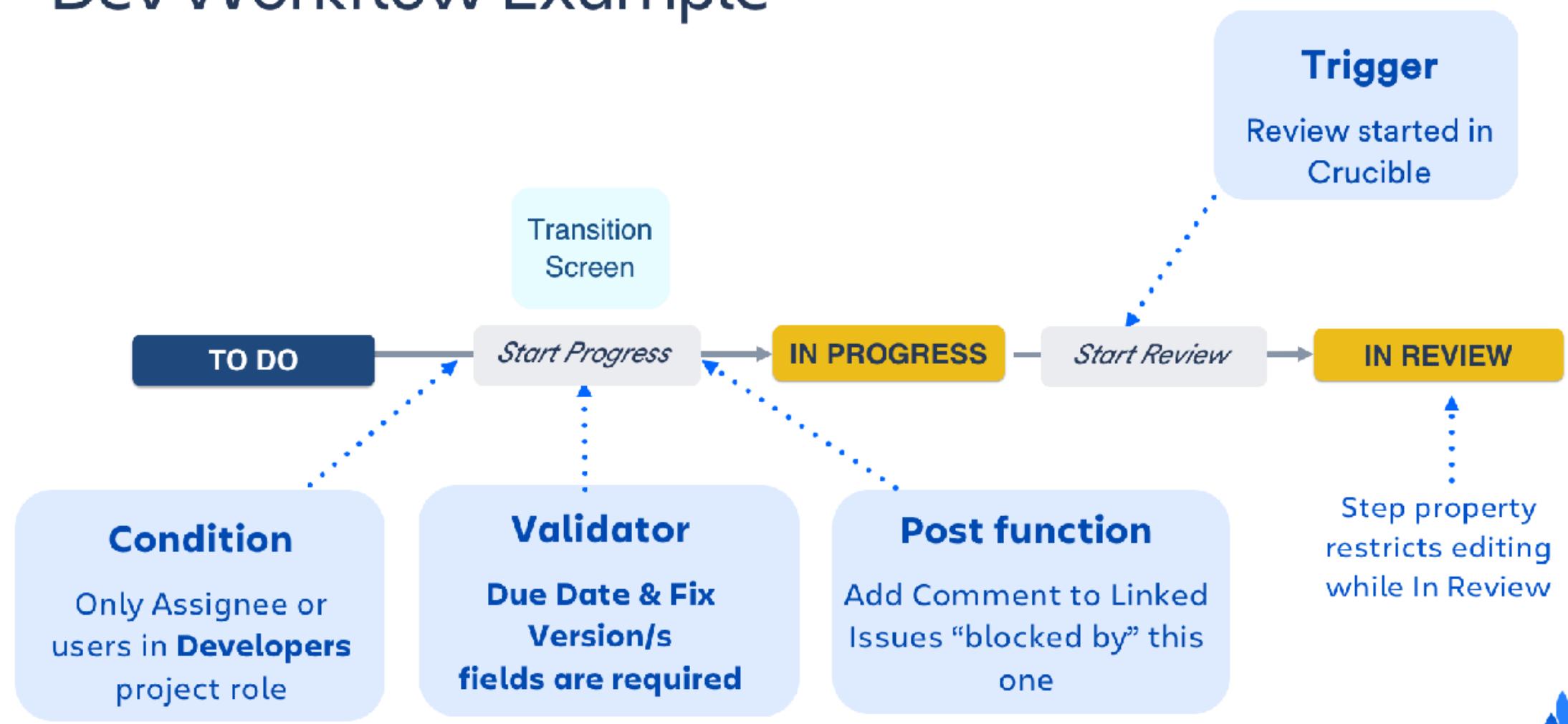
Editing and Testing

Common Problems & Troubleshooting

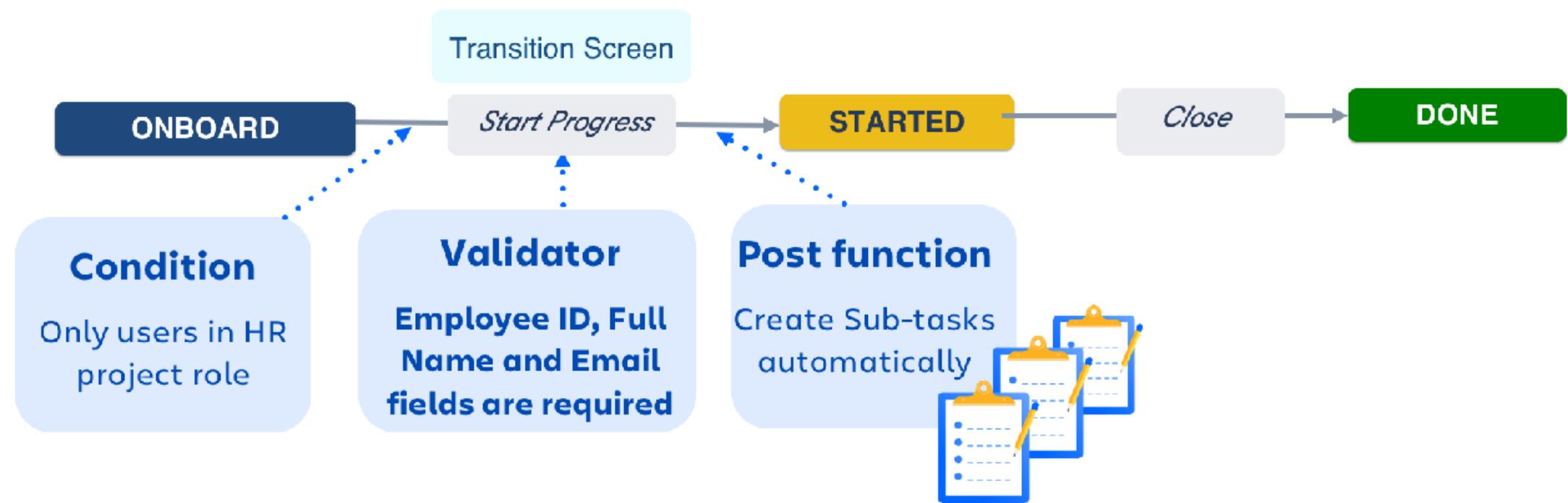
Best Practices



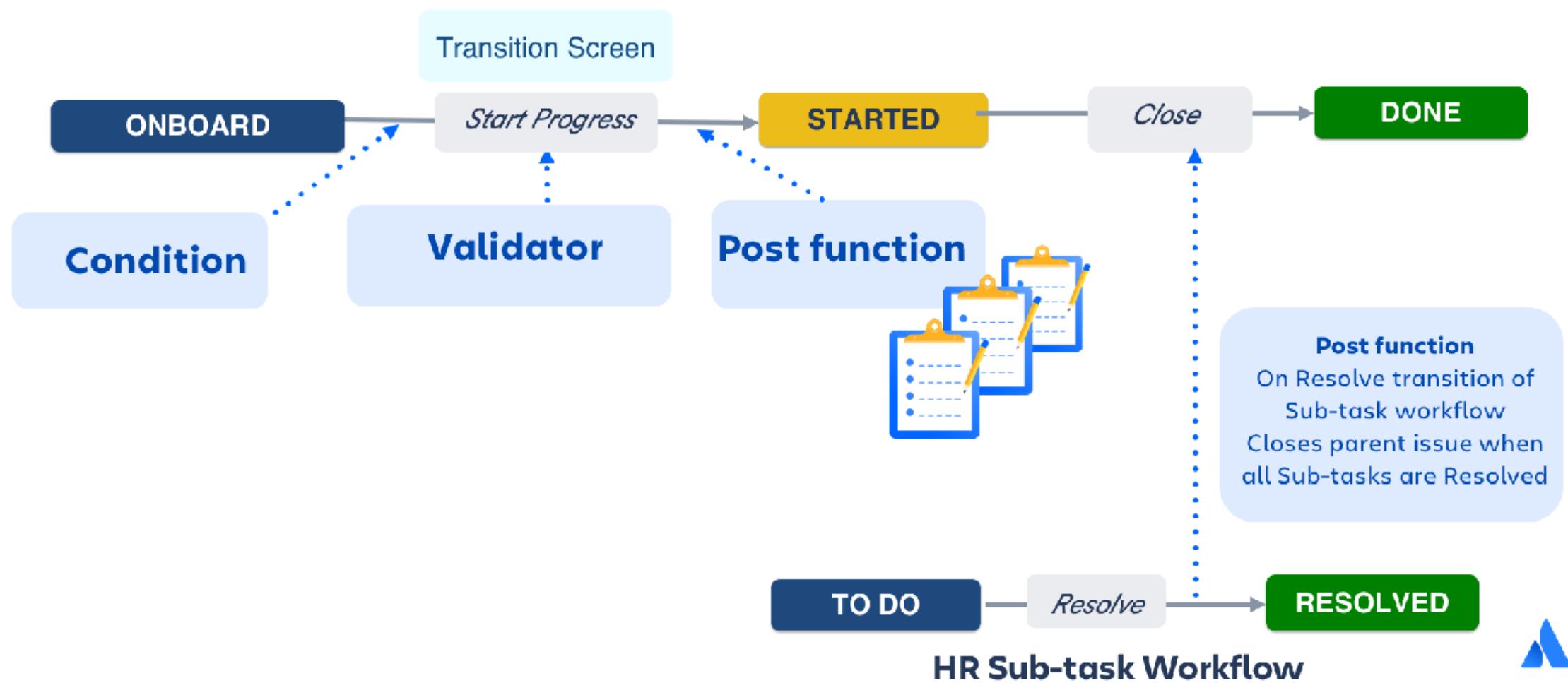
Dev Workflow Example



HR Workflow Example



HR Sub-task Workflow



REQUIREMENT



You have to change the Due Date while rescheduling a Task.

?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT



You have to change the Due Date **while rescheduling** a Task.

VALIDATOR



Field has been modified

Validators can check if the user has modified a field during the transition



REQUIREMENT

While closing a Ticket, the Root Cause & Comment should be copied to its Duplicates.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

While closing a Ticket, the Root Cause & Comment should be **copied to its Duplicates**.



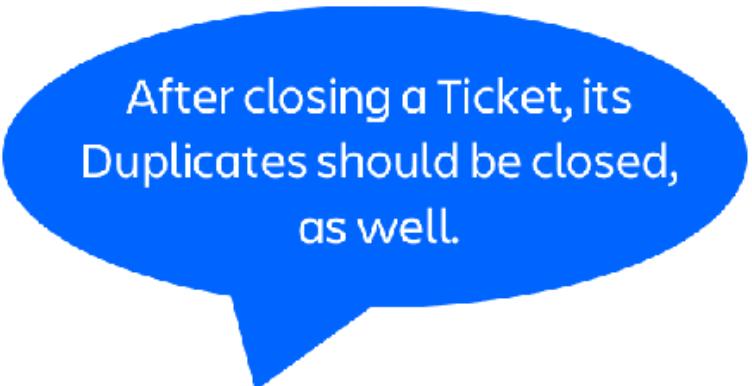
POST-FUNCTION



Copy field value or comment



REQUIREMENT



After closing a Ticket, its Duplicates should be closed, as well.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

After closing a Ticket, its Duplicates should be closed, as well.



POST-FUNCTION



Transition linked issue

Post Functions can transition current, linked, or parent issue to another status in its workflow



REQUIREMENT

Don't start the upgrade task until all the prerequisites are completed.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

Don't start the upgrade task
until all the prerequisites are completed.



CONDITION



Status of Linked Issues

Conditions can check if linked issues are in particular statuses



REQUIREMENT

Allow only Auditors to edit issues in the In Review status.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

Allow **only** Auditors to **edit** issues in the In Review **status**.



STEP PROPERTY



jira.permission.edit.group.1=Auditors

Step Properties can restrict certain actions, like **editing**, in a certain status, only to certain people.



REQUIREMENT



Only Testers can create Bugs in this project, but anyone can create Tasks.

?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT



Only Testers can create
Bugs in this project, but
anyone can create Tasks.

VALIDATOR



Boolean expression

Validators can limit issue
creation for a specific issue type
and project role in the same
workflow.



REQUIREMENT

When Tests fail, let's create new Bugs automatically.



?

CONDITION?

VALIDATOR?

POST FUNCTION?

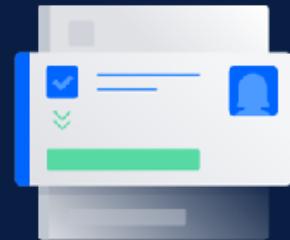


REQUIREMENT

When Tests fail, let's **create new Bugs** automatically.



POST-FUNCTION



Create new issue

Post Functions can create new related issue (linked issue, Sub-task, issue in Epic)



REQUIREMENT

Require purchase approval
only if purchase price exceeds
\$10,000.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

Require purchase approval
only if purchase price exceeds
\$10,000.



CONDITION



Field Value

Conditions can check if a system
or custom field is empty or has a
particular value



REQUIREMENT

Reporters should select only one component when creating Bugs.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT



Reporters should select **only one** component **when creating** Bugs.

VALIDATOR



Field has single value

Validators can make sure that user selected only one value from multi-value field during transition



REQUIREMENT

Only the requestor's manager
– as listed in the issue – can
approve this Change
Request.



?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

Only the requestor's manager
– as listed in the issue – **can**
approve this Change
Request.



CONDITION



User in Field

Conditions can check if the current user is present or absent in a single user or group user custom field



REQUIREMENT



Mark the Fix Version/s field as required, but only when the resolution is set to Fixed during the transition.

?

CONDITION?

VALIDATOR?

POST FUNCTION?



REQUIREMENT

Mark the Fix Version/s field as required, but only when the resolution is set to Fixed **during the transition**.



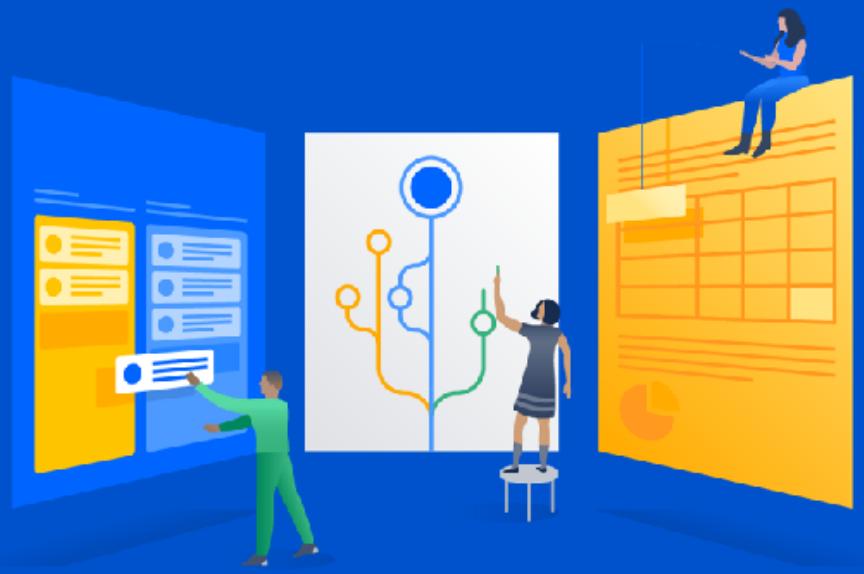
VALIDATOR



Behaviors

Behaviors define how fields behave during a transition, such as being required, shown/hidden, or having different options based on the value in another field.





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

Editing and Testing

Common Problems & Troubleshooting

Best Practices



Service Desk Workflow Permissions



To use workflows in Jira Service Desk projects, users need:

- **Browse Projects** permission – to see the project at all
- **Transition Issues** permission – to see any transitions
- Jira Service Desk Application access – to see any transition in Service Desk workflows (and other features like Queues, SLAs, etc.)
- Any other permission, groups, project roles, as required by conditions and validators, in addition to the above



Request Types Mapped to Issue Types

Common Requests				
Icon	Request name	Issue type	Description (Optional)	
		<input checked="" type="checkbox"/> Task		
	Get IT help	<input type="checkbox"/> Service Request	Get assistance for general IT problems and questions.	
	Set up VPN to the office	<input type="checkbox"/> Service Request	Want to access work stuff from outside? Let us know.	
	Report a system problem	<input type="checkbox"/> Incident	Having trouble with a system?	

Request Types
are mapped to
Issue Types



Request Types are mapped to Issues Types which are mapped to workflows.



Issue Types Mapped to Workflows

The screenshot shows the Jira Service Desk interface for managing request types and their mappings.

Request types: A list of request types grouped under "Common Requests".

Icon	Request name	Issue type	Description (Optional)
Monitor icon	Get IT help	Service Request	Get assistance for general IT problems and questions.
Padlock icon	Set up VPN to the office	Service Request	Want to access work stuff from outside? Let us know.
Warning icon	Report a system problem	Incident	Having trouble with a system?

Issue Types are mapped to Request Types

Issue Types are mapped to workflows in a workflow scheme

Workflow Scheme: A table showing the mapping of workflows to issue types.

Workflow	Issue Types
SDIT: Change Management workflow for Jira Service Desk. (View as text)	Change (Assign)
SDIT: Incident Management workflow for Jira Service Desk. (View as text)	Incident (Assign)
SDIT: Jira Service Desk default workflow. (View as text)	Task Sub-task (Assign)
SDIT: Problem Management workflow for Jira Service Desk. (View as text)	Problem (Assign)
SDIT: Service Request Fulfilment with Approvals workflow for Jira Service Desk. (View as text)	Service Request with Approvals (Assign)
SDIT: Service Request Fulfilment workflow for Jira Service Desk. (View as text)	Service Request (Assign)



Workflow Scheme Associated with Project

The screenshot shows the Jira Service Desk interface. On the left, under 'Request types', there's a 'Common Requests' group containing three items: 'Logins and Accounts', 'Computers', and 'Applications'. Below this is a section for 'Hidden from portal' requests, which is currently empty. In the center, there's a table titled 'Common Requests' with columns for 'Icon', 'Request name', 'Issue type', and 'Description (Optional)'. It lists three items: 'Get IT help' (Service Request, description: 'Get assistance for general IT problems and questions.'), 'Set up VPN to the office' (Service Request, description: 'Want to access work stuff from outside? Let us know.'), and 'Report a system problem' (Incident, description: 'Having trouble with a system?'). On the right, a 'Workflow Schemes' panel is shown, with a callout pointing to it from the top right. The panel title is 'Workflow Schemes are associated with projects'. It shows a list of workflow schemes: 'SDIT: Change Management workflow for Jira Service Desk' (Change (Assign)), 'SDIT: Incident Management workflow for Jira Service Desk' (Incident (Assign)), 'SDIT: Jira Service Desk default workflow' (Task, Sub-task (Assign)), 'SDIT: Problem Management workflow for Jira Service Desk' (Problem (Assign)), 'SDIT: Service Request Fulfilment with Approvals workflow for Jira Service Desk' (Service Request with Approvals (Assign)), and 'SDIT: Service Request Fulfilment workflow for Jira Service Desk' (Service Request (Assign)).

Request Types are mapped to Issue Types

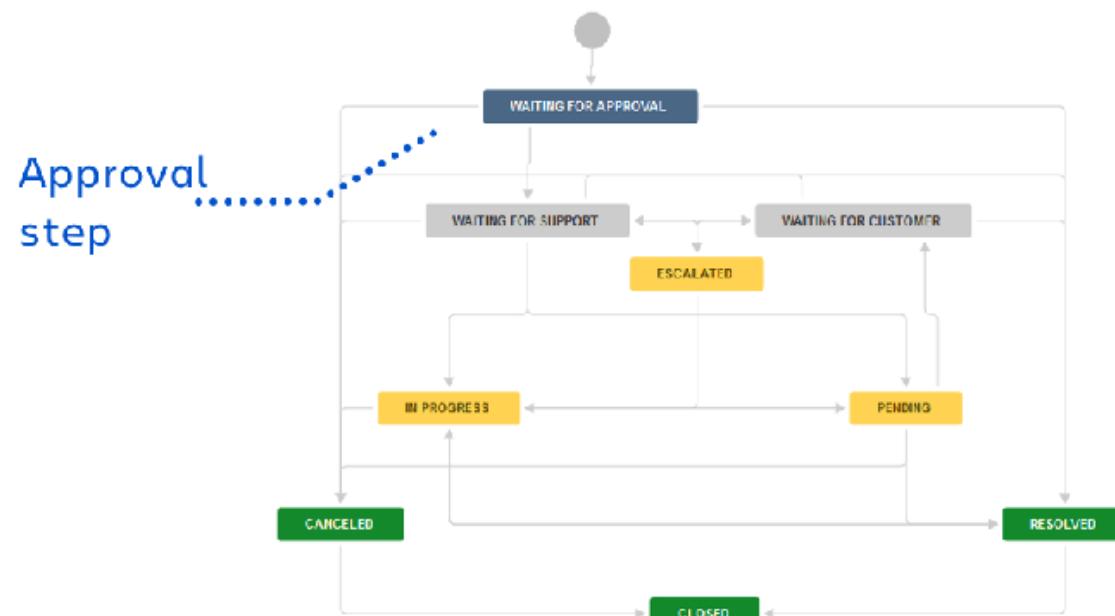
Issue Types are mapped to workflows in a workflow scheme

Workflow Schemes are associated with projects

Service Desk Approval Workflows

↳ Jira Service Desk

Service Request Fulfilment with Approvals workflow

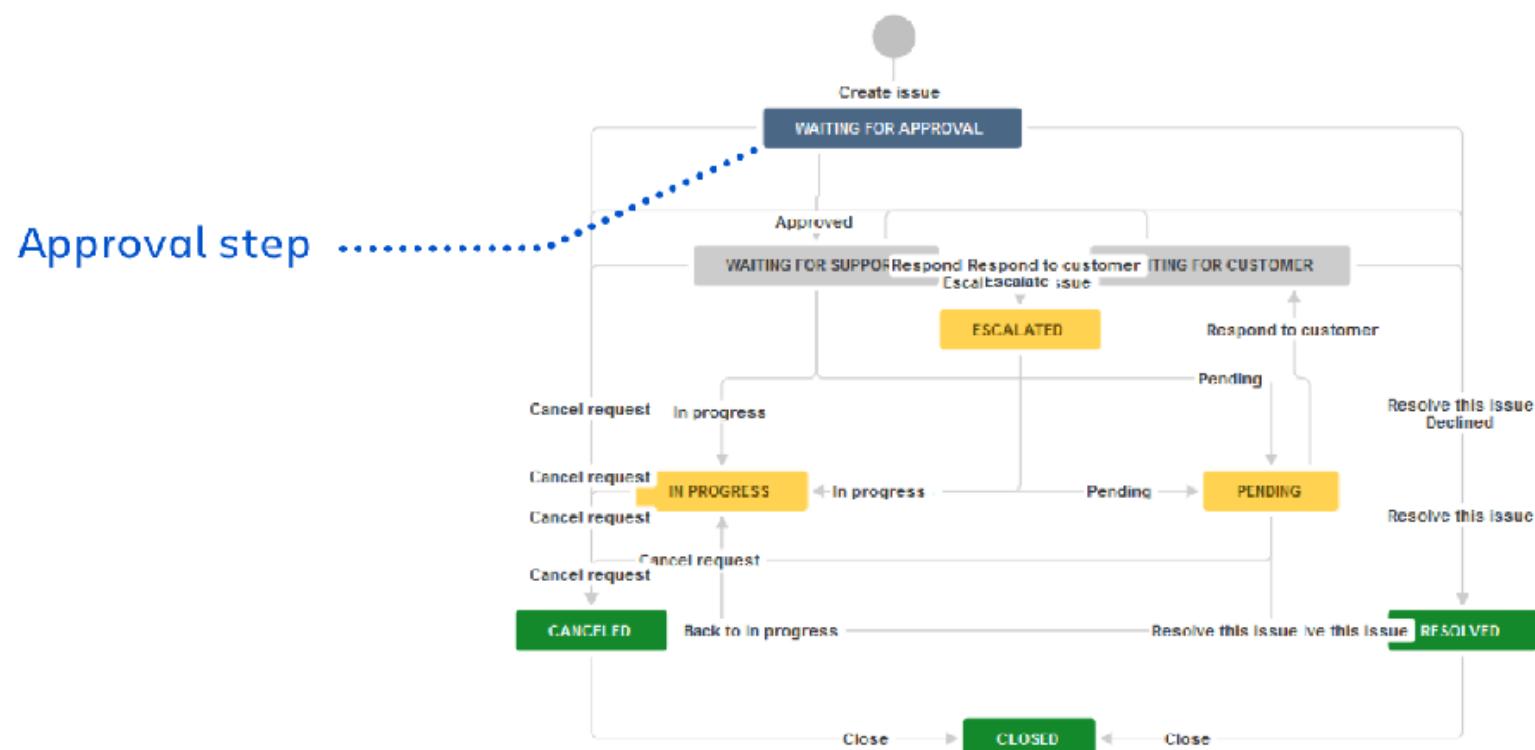


Change Management workflow

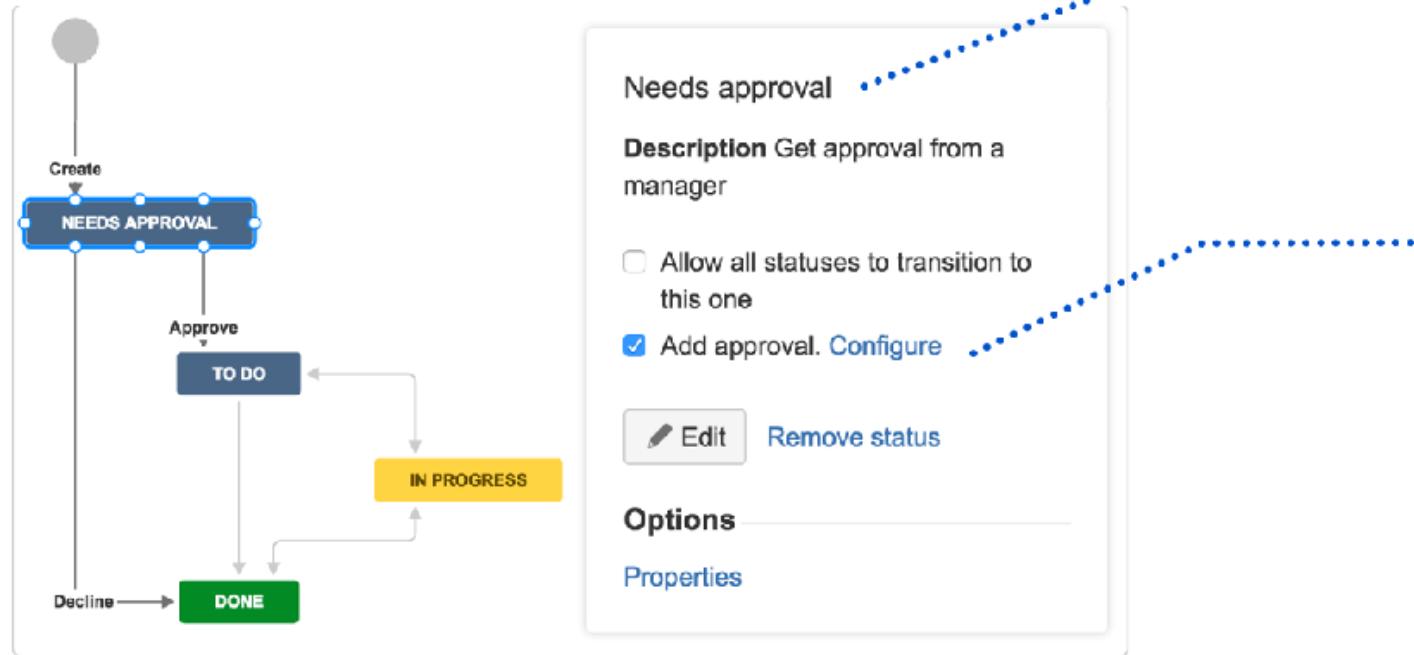


Service Desk Approvals

Service Request Fulfilment with Approvals workflow



Configuring the Approval Step



Select the Status where approval is needed

Click Add approval



Approval transitions don't display screens on the Customer Portal.



Configuring Approvers and Transitions

Add approval

Get approvers from: **Approvers**

Consider approved after: All approvals
 1 approval(s)

Transition when approved: **Approved**

Transition when declined: **Declined**

Create **Cancel**

Choose the User Picker field to use for list of Approvers
* Can create/use custom field

Choose required number of approvers

Choose approved and declined transitions

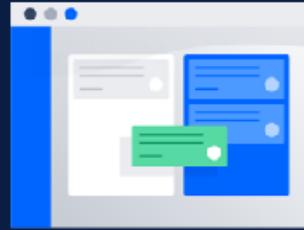


Multiple approval steps can be configured in a workflow.



Service Desk Automation Rules

Transition Issues



Send Alerts

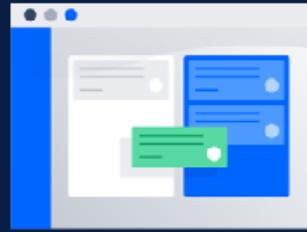


Update Issues



Service Desk Automation Rules

Transition Issues



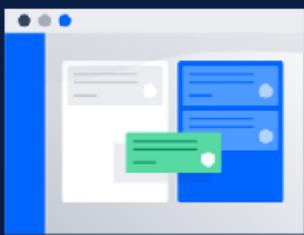
Transition on comment

Re-open on customer comment



Service Desk Automation Rules

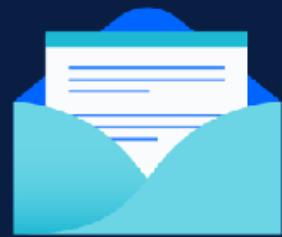
Transition Issues



Transition on comment

Re-open on customer comment

Send Alerts



Be aware of urgent issues

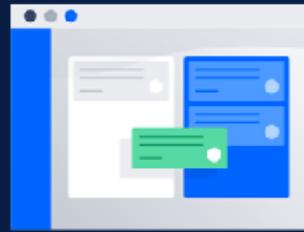
Keep on top of SLAs

Set customer expectations



Service Desk Automation Rules

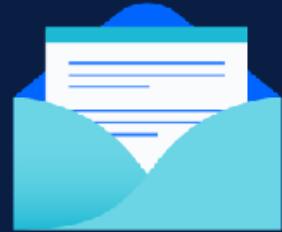
Transition Issues



Transition on comment

Re-open on customer comment

Send Alerts



Be aware of urgent issues

Keep on top of SLAs

Set customer expectations

Update Issues



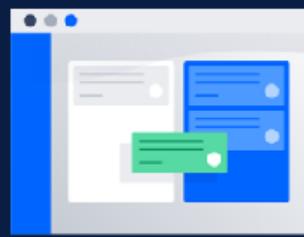
Update when linked issue changes

Triage requests sent by email



Service Desk Automation Rules

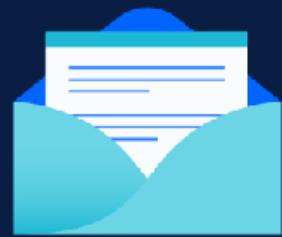
Transition Issues



Transition on comment

Re-open on customer comment

Send Alerts



Be aware of urgent issues

Keep on top of SLAs

Set customer expectations

Update Issues



Update when linked issue changes

Triage requests sent by email



Custom Rule allows you to create your own custom rule, as needed.



How Rules Work

WHEN

Triggers



- Comment added
- Comment edited
- Issue created
- Issue resolution changed
- Status changed
- A linked issue is transitioned
- Participant added
- Organizations added to issue
- Approval required
- SLA time remaining



How Rules Work

WHEN Triggers



- Comment added
- Comment edited
- Issue created
- Issue resolution changed
- Status changed
- A linked issue is transitioned
- Participant added
- Organizations added to issue
- Approval required
- SLA time remaining

IF (optional) Conditions



- Issue matches
- User type



How Rules Work

WHEN Triggers



- Comment added
- Comment edited
- Issue created
- Issue resolution changed
- Status changed
- A linked issue is transitioned
- Participant added
- Organizations added to issue
- Approval required
- SLA time remaining

IF (optional) Conditions



- Issue matches
- User type

THEN Actions



- Transition issue
- Add comment
- Alert user
- Edit request type
- Edit issue
- Send email
- Webhook
- Auto-approve/decline



Who Configures



Jira Administrators or
Project Administrators
can configure rules

Who Runs



Rules run as project
default
or as the user who
triggered the rule



Automation vs. Workflow Customization

- It occurs behind-the-scenes without need for user intervention



Automation vs. Workflow Customization

- It occurs behind-the-scenes without need for user intervention
- Project Administrators have more control

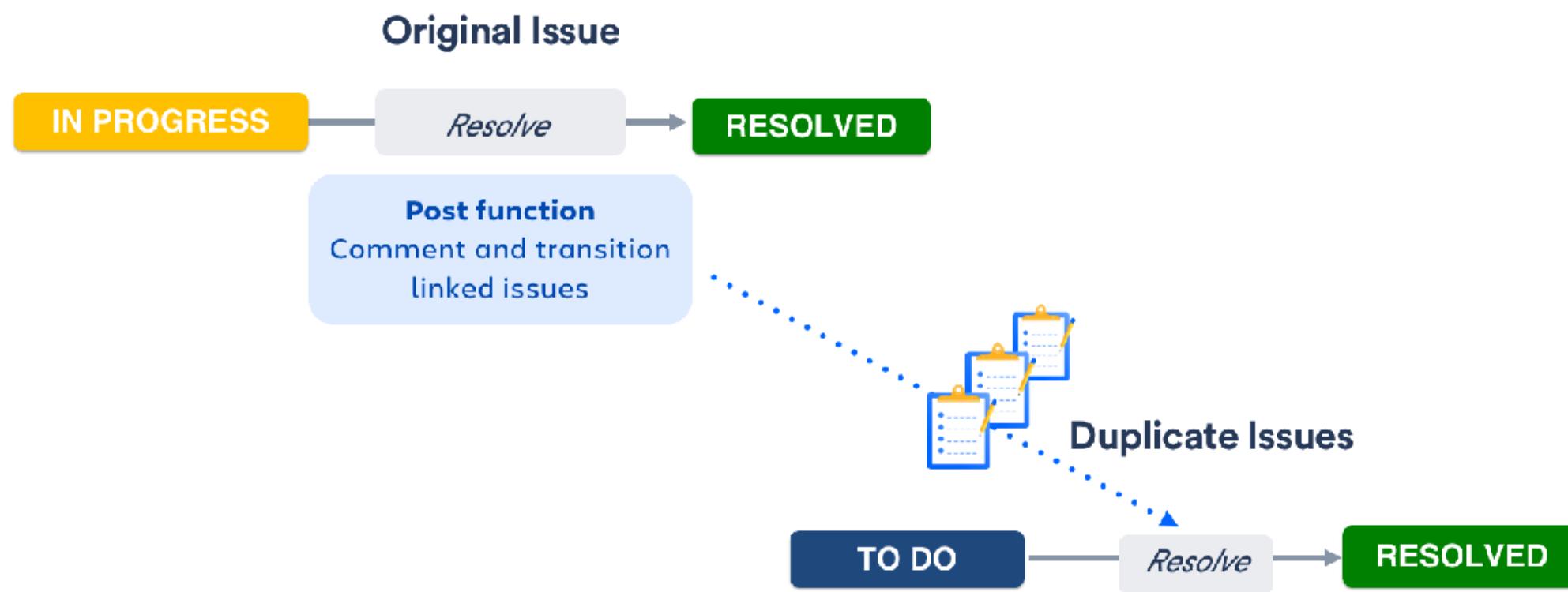


Automation vs. Workflow Customization

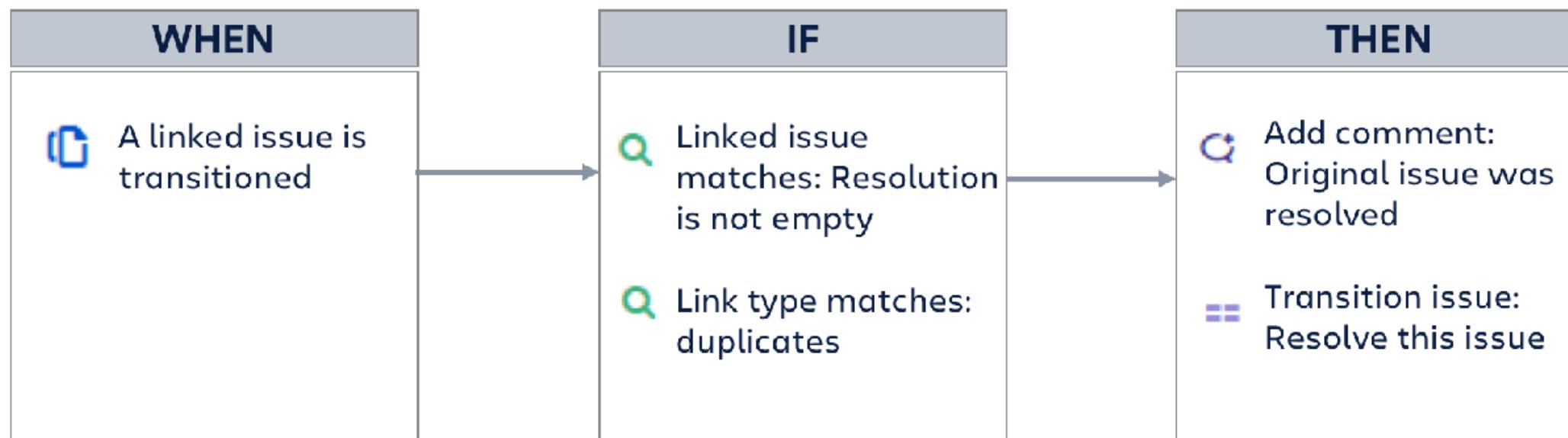
- It occurs behind-the-scenes without need for user intervention
- Project Administrators have more control
- Automation is easier to configure and maintain
- You can use both



Workflow Customization



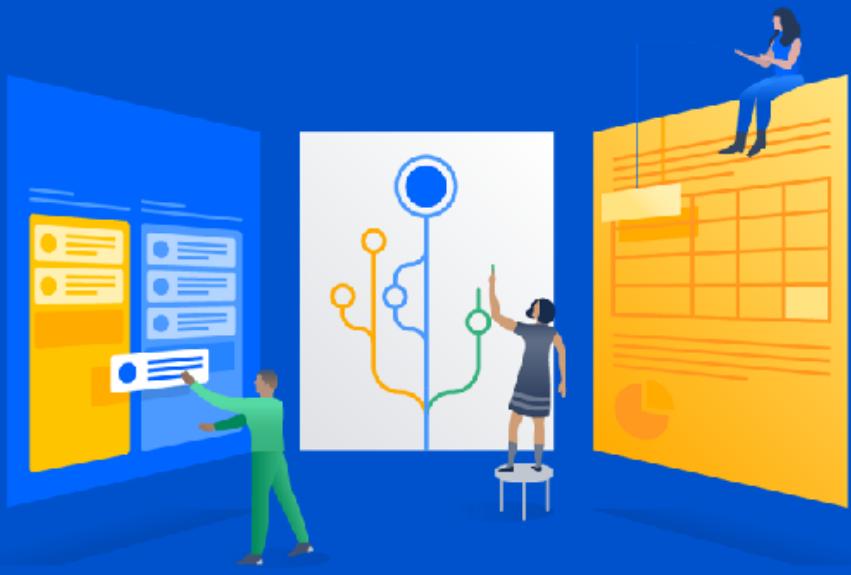
Automation Rule



Automation Best Practices

- Ensure the person configured to run rules has appropriate permissions
- Beware of rules triggering each other
- Use native Service Desk automation functionality whenever possible
- Test and evaluate in staging first





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

Editing and Testing

Common Problems & Troubleshooting

Best Practices



Who Can Edit Workflows?



Jira Administrator Has Full Edit Rights



Jira Administrator

Full access to edit workflows
and workflow schemes and
associate with projects



Project Administrator Has Limited Rights



Project Administrator

Limited workflow editing with
“Extended Project
Administration” permission



Extended Project Administration

Project permissions

Permission	Granted to
Administer Projects	Project role
Ability to administer a project in JIRA.	<ul style="list-style-type: none">Administrators
<input checked="" type="checkbox"/> Extended project administration Grant extended project administration permissions .	

Required for project administrators to edit their workflows (and screens)



PROJECT ADMINS CAN

- Edit the workflow associated with their project (if unshared)
- Create, update, delete transitions
- Add existing statuses to their workflow
- Remove statuses that aren't used by their project's issues

PROJECT ADMINS CAN'T

- Edit shared workflows
- Edit the Jira default system workflow
- Select or update a transition screen
- Edit transition properties, conditions, validators, or post functions
- Create new statuses
- Change workflow schemes



Board Administrator Might Be Able to Add Statuses

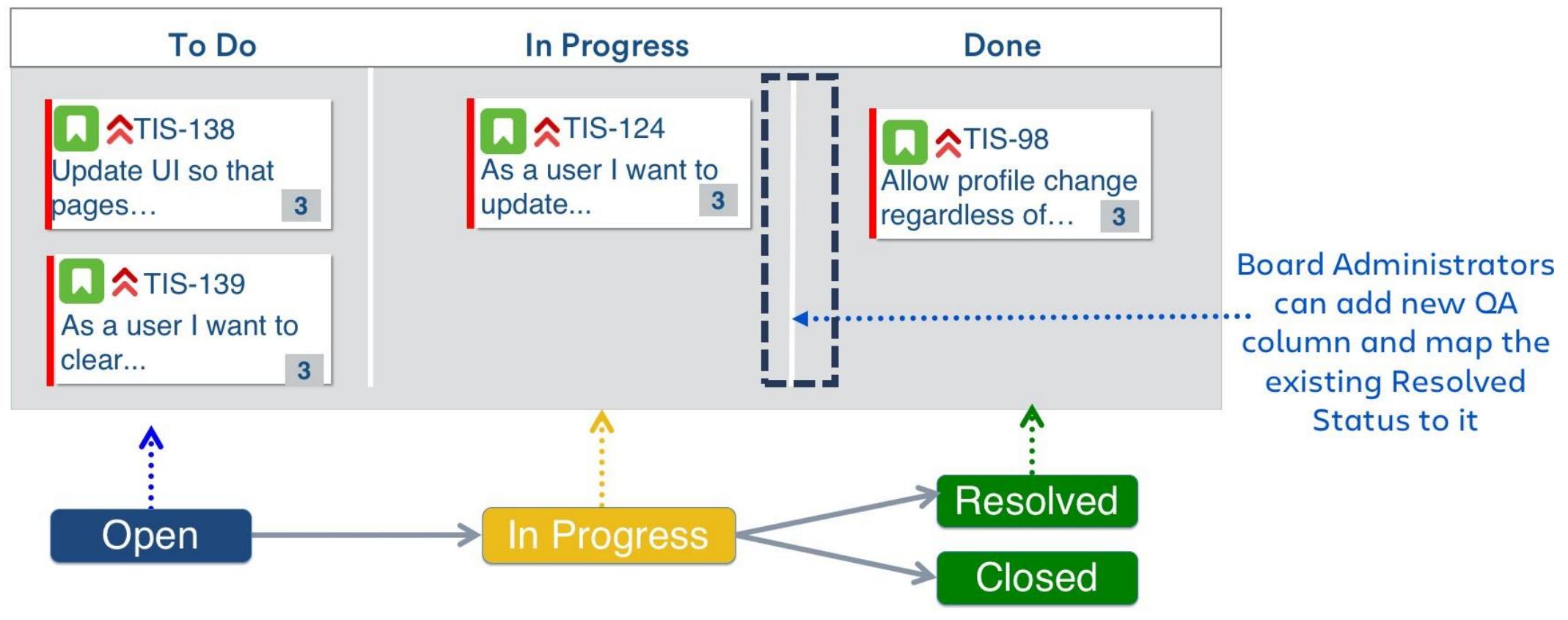


Board Administrator
who is also a Project Administrator

Add a Status to a Board (&
hence to the workflow). Board
must use Simplified Workflow



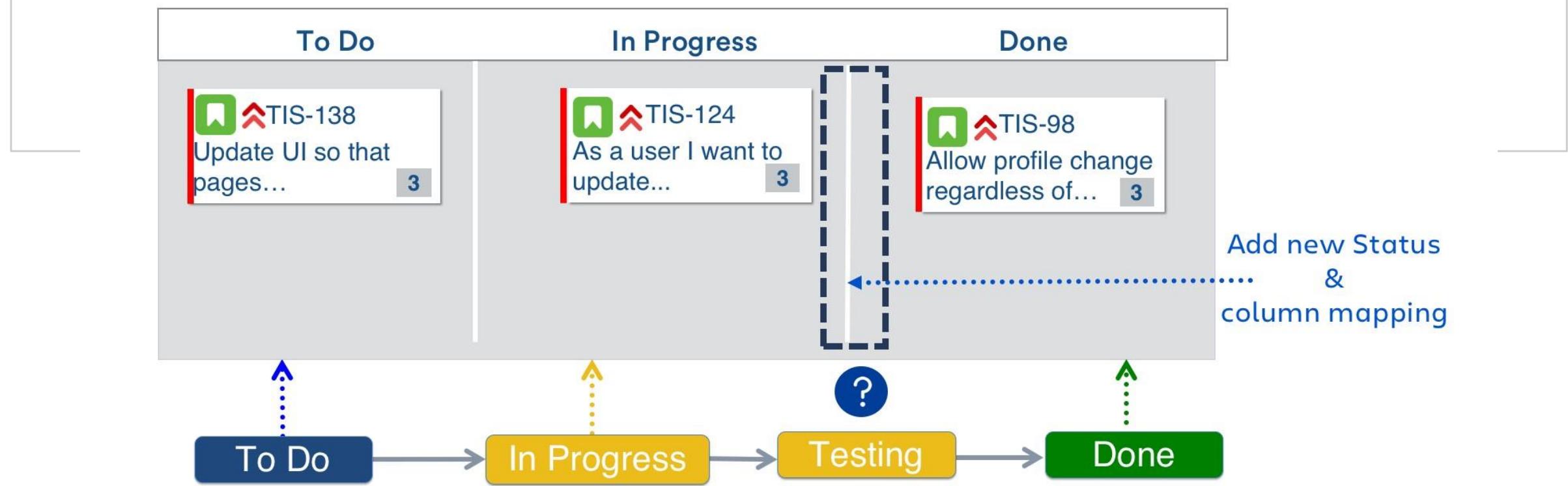
Modifying Board Columns



Board Administrators can change board column mapping.



Adding Statuses to Boards



- i** Board Administrators, who are also Project Administrators can add Statuses if the board is using the Simplified Workflow.



Adding Statuses to Boards

Configure DEV board

Back to board Board

CONFIGURATION

General

Columns

Swimlanes

Quick Filters

Card colours

Card layout

Estimation

Working days

Issue Detail View

Column management

Columns can be added, removed, reordered and renamed. Columns are based upon global statuses and can be moved between columns. Minimum and maximum constraints can be set for each mapped column.

Column Constraint: None

Constraints can be added to columns on the board for one statistic.

Days in column: Shows dots representing the time an issue spent in a particular column on Jira board.

Simplified Workflow

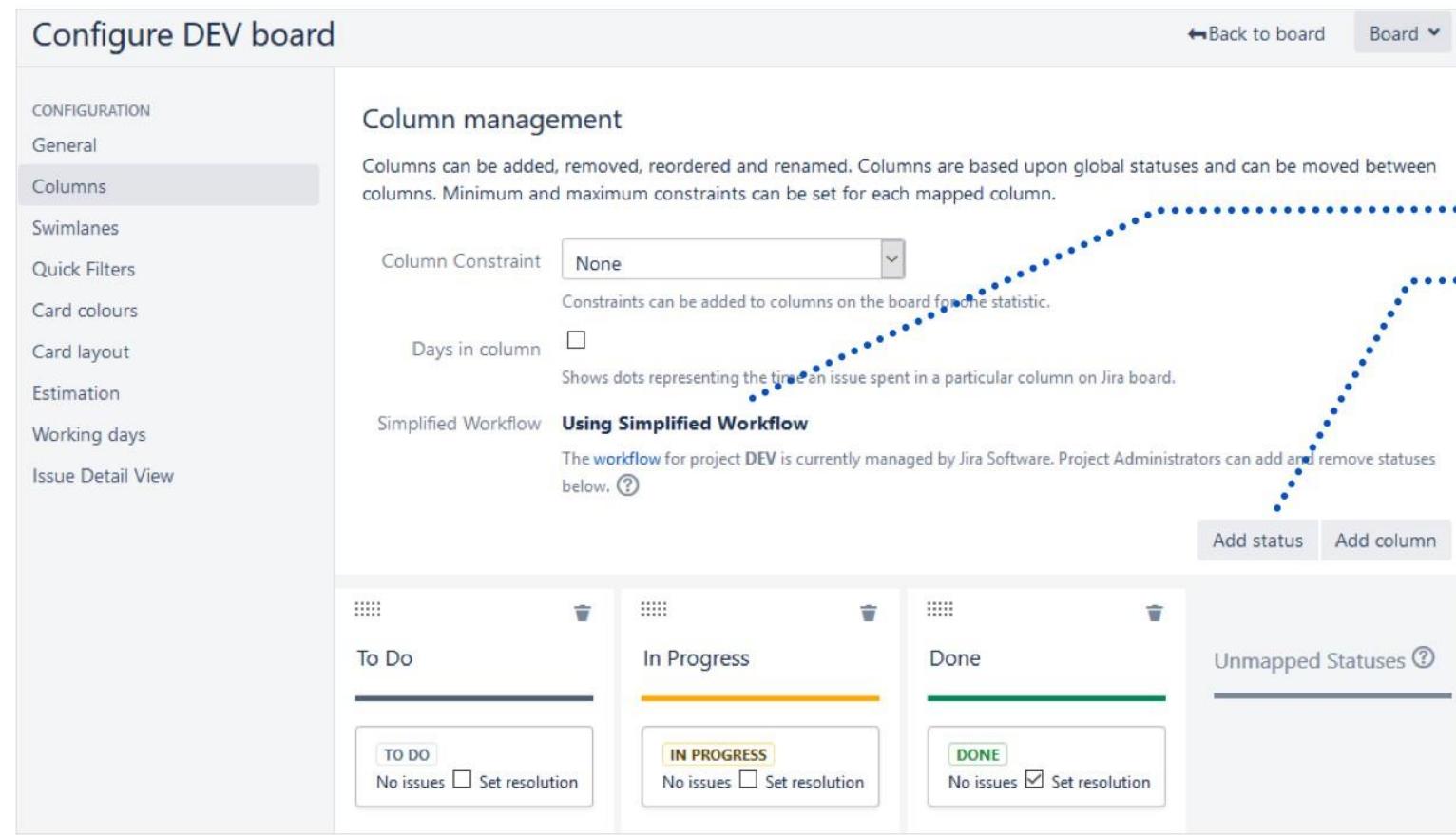
The workflow for project DEV is currently managed by Jira Software. Project Administrators can add and remove statuses below. [?](#)

Add status Add column

To Do In Progress Done Unmapped Statuses [?](#)

TO DO IN PROGRESS DONE

No issues Set resolution No issues Set resolution No issues Set resolution



To add a status to a board, the project must be using Simplified Workflow. The workflow cannot be shared.



Be watchful of users creating a lot of statuses



Who Can Edit Workflows?



Jira Administrator



Project Administrator



Board Administrator
who is also a Project Administrator

Full access to edit workflows
and workflow schemes and
associate with projects

Limited workflow editing with
“Extended Project
Administration” permission

Add a Status to a Board (&
hence to the workflow). Board
must use Simplified Workflow



Switching to Simplified Workflow

- Only one project being viewed by the board
- Only one workflow for all issue types
- Atlassian-provided conditions, validators, post functions
- At least one outgoing transition for each status



Building New Workflows

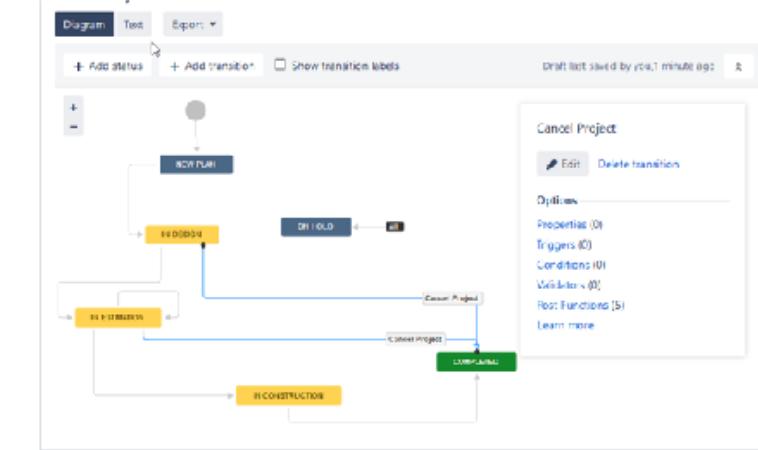
- Copy and customize an existing workflow rather than creating from scratch
- Use a small set of standard workflows
- Start simple; add on when requirements warrant



Editing Workflows

- Edit Active workflows by creating a draft and publishing the draft when ready
- Use Text or Diagram modes; finalize diagram for end-users
- An Inactive workflow is not currently being used by any projects; it must be activated
- Activating means mapping an Inactive workflow to a scheme and associating to a project

Diagram Mode



Text mode

Step Name (id)	Linked Status	Transitions (id)	Actions
New Plan (2)	NEW PLAN	Start Design (51) >> [IN DESIGN] Put On Hold (11) >> On Hold	Add transition Delete Transitions Edit View Properties
In Estimation (3)	IN ESTIMATION	Reestimate (41) >> [IN ESTIMATION] Start Construction (61) >> [UNDER REVIEW] Cancel Project (81) >> [COMPLETED] Put On Hold (11) >> On Hold	Add transition Delete Transitions Edit View Properties



Limitations on Editing Active Workflows

- Can only edit the description (not name) of an active workflow
- Workflow steps cannot be deleted
- The associated Status of a step cannot be edited
- Step ID cannot be changed
- If a step has no outgoing transitions (global transitions are not considered), it cannot have any new outgoing transitions added



Working With Statuses

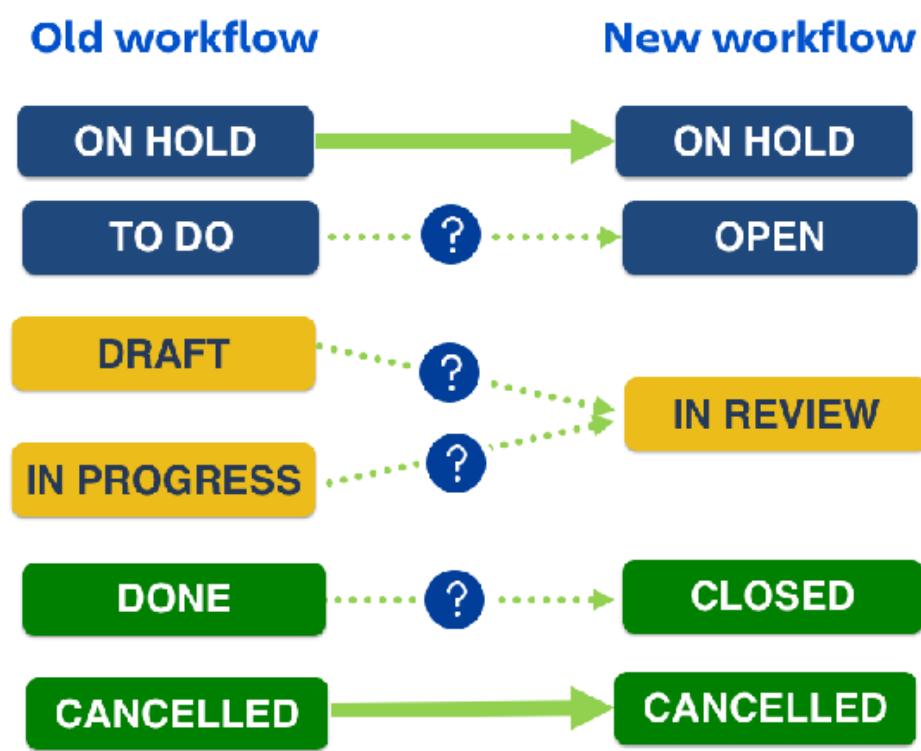
- Changing name of a status on one workflow also changes it in all workflows that use that status
- Check before removing a status that it's not being used in other places
- Removing statuses causes issue migration when the modified workflow is associated with project

Statuses in Jira

Statuses					
Name	Category	Workflows	Order	Actions	
Open	To Do	9 associated workflows	↓	Edit	
In Construction	In Progress	14 associated workflows	↑ ↓	Edit	
Reopened	To Do	2 associated workflows	↑ ↓	Edit	
Resolved	Done	2 associated workflows	↑ ↓	Edit	
Closed	Done	0 associated workflows	↑ ↓	Edit	



Migrating Issues



Workflow migrations can take a long time.



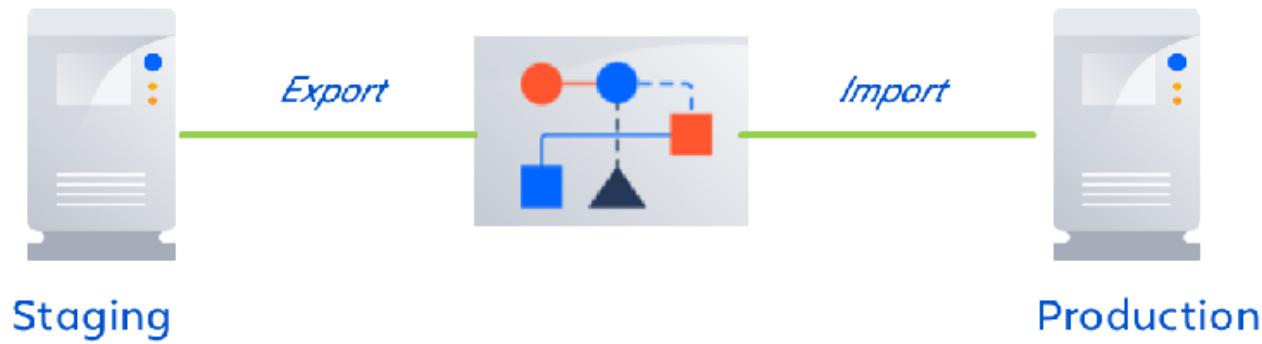
Use a Test Jira Instance

- Best to build and test workflows robustly on a test instance
- Use a test project on a production instance
- Use caution when editing and publishing the draft of an active workflow
- Save a backup copy when publishing in production



Workflow Import / Export

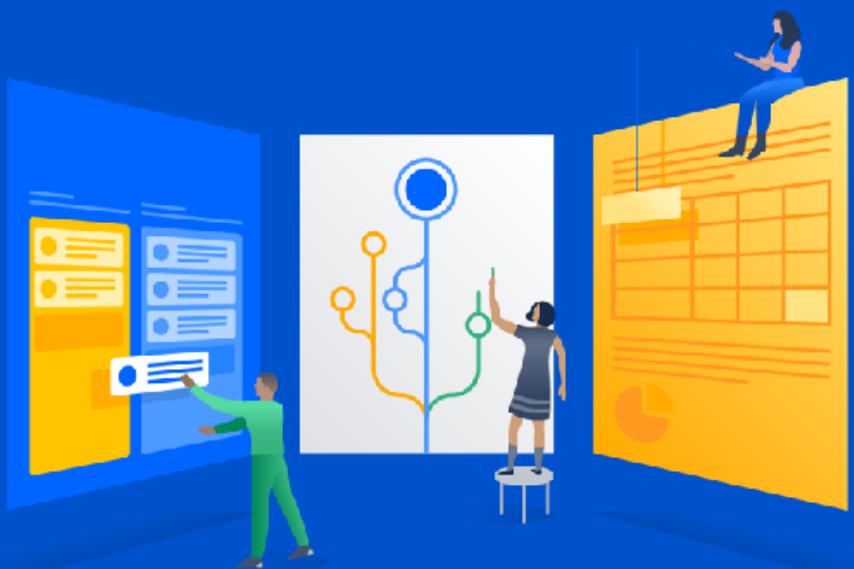
- Import/Export as XML or Workflow between instances or share on Marketplace
- Not all workflow functionality may be exported/imported
- Importing XML workflows is supported for Server only



Test Everything

- Go through each status and transition
 - ✓ Check all Conditions, Validators, Transition Screens
 - ✓ See if all Post Functions behave as expected
 - ✓ Check Triggers and connected Development Tools
 - ✓ Check if the right event was fired
 - ✓ Check if the right notifications were sent
- Test users, permissions, memberships in groups and project roles





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

Editing and Testing

Common Problems & Troubleshooting

Best Practices



Common Problems

Troubleshooting
Basics

Permissions

Misconfiguration

Administrator
Problems

- Try to replicate the problem
- Open the workflow in the workflow editor
- Check all statuses and transitions
- Check conditions, validators, post functions, triggers
- Check properties
- Check automation rules and approval functionality in Jira Service Desk
- View Audit Log



Common Problems

Troubleshooting
Basics

Permissions

Misconfiguration

Administrator
Problems

- User cannot see any transitions (check Transition Issues permission, and possibly Application Access)
- User cannot see one transition (check permissions or project roles in Condition)
- User cannot drag card to a particular column on a board (transition condition is not met in underlying workflow)
- No one can edit Closed issues (check Step Property: `jira.issue.editable`)
- Some users cannot comment in a particular status (check Step Property: `jira.permission.comment.*`)



Common Problems

Troubleshooting
Basics

Permissions

Misconfiguration

Administrator
Problems

- User cannot add / update a required field during transition (field may not be on transition screen)
- User is not notified when a transition occurs (wrong event fired or user not listed as event recipient in notification scheme)
- Saved filter or gadget returning data errors (workflow status was possibly removed or renamed)



Common Problems

Troubleshooting

Basics

Permissions

Misconfiguration

Administrator
Problems

- Resolved issues showing as Open (workflow not setting Resolution field)
- Open issues showing as Resolved (workflow not clearing Resolution field)
- After editing workflow, other projects and issue types were affected (workflow was shared)
- Board column shows no issues (status was removed from workflow but column was not)
- Workflow returning strange errors in the logs (workflow app may be causing issues)





Workflow Basics

Default Workflows & Customizing

Configuring Transitions

Advanced Examples

Service Desk Approvals & Automation

Editing and Testing

Common Problems & Troubleshooting

Best Practices



Best Practices

General Configuration

Statuses

Transitions

- Check if workflow is shared before making changes to ensure your customizations won't impact other projects
- Explore Marketplace apps to extend workflow functionality
- Use native, out-of-the-box, Jira functionality & features where possible
- Enable Extended Project Administration so project admins can help maintain their workflows
- Document workflows
- Make your workflow diagram easy to understand



Best Practices

General Configuration

Statuses

Transitions

- Add statuses only when you need them
- Understand and limit who can add new statuses
- Use intuitive, more generic & reusable names
- Don't create statuses that don't transition out, unless it is the final status and should not allow reopening
- Ensure all statuses are mapped appropriately to columns on agile boards



Best Practices

General Configuration

Statuses

Transitions

- Add transitions only when needed
- Use intuitive names
- Use common and global transitions
- Ensure post functions are in the correct order
- Ensure each workflow sets and clears the Resolution appropriately
- Use project roles in conditions and validators

