Transition parent issue only when certain issue sub-task types are done

Features used to implement the example

- **Validation on sub-tasks**
- **Copy parsed text to a field**

Example: Transition parent issue only when certain issue sub-task types are done

User Stories have a user story workflow. The Engineering Sub-Task type has an engineering sub-task workflow. The QA Sub-Task type has a qa sub-task workflow.

I have the following compound criteria to meet:

1. When at least one Engineering Sub-Task or QA Sub-Task type is set to "**In Progress**", auto transition the **parent User Story** to **Step 2**.
2. Only when all Engineering Sub-Task types are "**Done**", auto transition the **parent User Story** to **Step 3**. This happens regardless of the status of the QA Sub-Tasks.
3. Only if all Engineering Sub-Task types and QA Sub-Task types are "**Done**", allow transition of the User Story from **Step 3 to Step 4**.

I've seen a few workflow plugins that allow transition of the parent if *all* sub-task types are set to a particular status or resolution. That would violate #2, however. Are there any plugins that allow you to segregate sub-task types for different transition thresholds? I've been searching and thus far I haven't found any.
From a logic perspective it could work like this:

- Post Function on the Engineering Sub-Task workflow for the "**Done**" transition:
  - If all sibling Engineering Sub-Task types are "**Done**", auto-transition the parent issue from Step 2 to Step 3 (#2 in the compound criteria I listed above)
  - If all sibling Engineering Sub-Task types are "**Done**" and QA Sub-Task types are "**Done**", allow transition from Step 3 to Step 4.

Note: There is no allowed transition of the story from Step 2 to Step 4. That is handled at the User Story level.

Your requirements can be implemented in **Jira Workflow Toolbox** using two features of this plugin:

- **Validation on sub-tasks**: to be inserted in "**From Step 2 to Step 3**" and "**From Step 3 to Step 4**" transitions of Story Workflow, in order to prevent transition ejecutions if criteria 2 and 3 are not met. This validators work always, i.e. in human triggered transitions or post-function triggered transitions.

- **Copy parsed text to a field** to write on virtual fields "**Parent's issue status**" and "**Parent's issue status (delayed writing)**": when you write the name of a status into these virtual fields (provided by the plugin) you are transitioning parent's issue to that status, provided there is a transition in parent's workflow from current status to desired status, and all conditions and validators of that transition are met. The difference between them is that "**Parent's issue status (delayed writing)**" waits for sub-task transition to be executed, so that it is in the new status before parent transition is attempt. This way validator **Validation on sub-tasks** won't block parent transitioning if triggering sub-task was the lasting **Engineering Sub-Task** to be moved to "**Done"** status.

Now I explain in detail how to do it. I'm considering that "**Done**" is a status in sub-tasks workflows, but if it were a Resolution value it wouldn't be a problem.

Post-function Copy parsed text to a field in transition "Start Progress" in "Engineering Sub-Task" and "QA Sub-Task" workflows

This post-function will auto-transition parent issue to "**Step 2**" when any of the sub-tasks are moved to status "**In Progress**". Use the configuration shown in the screenshot. Notice that "**Status of Step 2**" must be replaced with the name of the status associated with Step 2:

Target field:

Parent's issue status ⬍

Text to be parsed and then copied to target field:

Status of Step 2

Post-function Copy parsed text to a field in transition to status "Done" in "Engineering Sub-Task" workflow

This post-function will auto-transition parent issue to "**Step 3**" when the last **Engineering Sub-task** is moved to status "**Done**". This post-function work as a team with validator **Validation on sub-tasks** in transition "**From Step 2 to Step 3**" in Story workflow. Use the configuration shown in the screenshot. Notice that "**Status of Step 3**" must be replaced with the name of the status associated with Step 3:

Target field:

Parent's issue status (delayed writing) ⬍

Text to be parsed and then copied to target field:

Status of Step 3

Validator Validation on sub-tasks in transition "From Step 2 to Step 3" in Story workflow

This validator will prevent execution of transition from Step 2 to Step 3 unless every **Engineering Sub-task** is in status "**Done**":

Allowed subtask types:	☑ 🗖 Engineering Subtask
	☐ 🗖 QA Subtask
	☐ 🗖 Sub-task
	☐ ⊙ gh.issue.task
	Selected issue types will be allowed. Nevertheless, if you didn't select any issue type, and checked "*Allow unselected issue types*", there won't be applied any filter by issue type, i.e., every issue type will be allowed.

Allowed statuses:	☐ →╖ Open
	☐ ⟳ In Progress
	☐ ←╖ Reopened
	☐ ➔ Resolved
	☐ ⤴ Closed
	☐ →╖ Assigned
	☑ ♦ Done
	Subtasks in selected statuses will be allowed. Nevertheless, if you didn't select any issue status, and checked "*Allow unselected statuses*", there won't be applied any filter by status, i.e., every issue status will be allowed.

Allowed resolutions:	☐ **UNRESOLVED**, i.e. no resolution value
	☐ Fixed
	☐ Won't Fix
	☐ Duplicate
	☐ Incomplete
	☐ Cannot Reproduce
	Subtasks with selected resolutions will be allowed. Nevertheless, if you didn't select any issue resolution, and checked "*Allow unselected resolutions*", there won't be applied any filter by resolution, i.e., every resolution will be allowed.

| Minimum required number of subtasks: | 0 |
| | Minimum number of subtasks requiered fulfilling selected filtering conditions (issue type, status and resolution). |

| Maximum allowed number of subtasks: | 1000 |
| | Maximum number of subtasks allowed fulfilling selected filtering conditions (issue type, status and resolution). |

| Allow unselected issue types: | ☑ |
| | Subtasks in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, or number. Nevertheless, if none of the issue types are selected, checking this option will make the validator behave as if you had selected every issue type. |

| Allow unselected statuses: | ☐ |
| | Subtasks in unselected statuses will be ignored, i.e., they will be allowed regardless of their resolution, or number. Nevertheless, if none of the statuses are selected, checking this option will make the validator behave as if you had selected every status. |

| Allow unselected resolutions: | ☑ |
| | Subtasks in unselected resolutions will be ignored, i.e., they will be allowed regardless of their number. Nevertheless, if none of the resolutions are selected, checking this option will make the validator behave as if you had selected every resolution. |

| Message to show when validation fails: | Every Engineering Subtask must be in "Done" status to execute this transition. |

Update   Cancel

Validator Validation on sub-tasks in transition "From Step 3 to Step 4" in Story workflow

This validator will prevent execution of transition from Step 3 to Step 4 unless every **Engineering Sub-task** and **QA Sub-task** is in status "**Done**":

Allowed subtask types:	☑ Engineering Subtask
	☑ QA Subtask
	☐ Sub-task
	☐ gh.issue.task
	Selected issue types will be allowed. Nevertheless, if you didn't select any issue type, and checked *"Allow unselected issue types"*, there won't be applied any filter by issue type, i.e., every issue type will be allowed.

Allowed statuses:	☐ Open
	☐ In Progress
	☐ Reopened
	☐ Resolved
	☐ Closed
	☐ Assigned
	☑ Done
	Subtasks in selected statuses will be allowed. Nevertheless, if you didn't select any issue status, and checked *"Allow unselected statuses"*, there won't be applied any filter by status, i.e., every issue status will be allowed.

Allowed resolutions:	☐ ***UNRESOLVED***, i.e. no resolution value
	☐ Fixed
	☐ Won't Fix
	☐ Duplicate
	☐ Incomplete
	☐ Cannot Reproduce
	Subtasks with selected resolutions will be allowed. Nevertheless, if you didn't select any issue resolution, and checked *"Allow unselected resolutions"*, there won't be applied any filter by resolution, i.e., every resolution will be allowed.

| **Minimum required number of subtasks:** | 0 |
| | Minimum number of subtasks requiered fulfilling selected filtering conditions (issue type, status and resolution). |

| **Maximum allowed number of subtasks:** | 1000 |
| | Maximum number of subtasks allowed fulfilling selected filtering conditions (issue type, status and resolution). |

| **Allow unselected issue types:** | ☑ |
| | Subtasks in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, or number. Nevertheless, if none of the issue types are selected, checking this option will make the validator behave as if you had selected every issue type. |

| **Allow unselected statuses:** | ☐ |
| | Subtasks in unselected statuses will be ignored, i.e., they will be allowed regardless of their resolution, or number. Nevertheless, if none of the statuses are selected, checking this option will make the validator behave as if you had selected every status. |

| **Allow unselected resolutions:** | ☑ |
| | Subtasks in unselected resolutions will be ignored, i.e., they will be allowed regardless of their number. Nevertheless, if none of the resolutions are selected, checking this option will make the validator behave as if you had selected every resolution. |

| **Message to show when validation fails:** | Every Engineering and QA subtask must be in "Done" status to execute this transition. |

Update   Cancel

Other examples of that functions   Related Usage Examples