Automatically resolve an epic when all its stories are resolved

On this page

- Features used to implement the example
- Example: Automatically resolve an epic when all its stories are resolved
- Other examples of that functions
- Related Usage Examples

Features used to implement the example

- **Write field on linked issues or sub-tasks**
- **Block or hide a transition for an issue depending on its issue links**

We add **Write field on linked issues or sub-tasks** to "**Resolve Issue**" transition on Story's workflow, in order to execute transition "*Resolved Issue*" on Epic issue. To do it we write the name of the transition into virtual field "**Execute transition (delayed execution)**". We use a boolean expression in order to check that the rest of the stories are already resolved or closed.

We add **Blocking or hiding a transition depending on its issue links** to "**Resolve Issue**" transition in **Epic's** workflow, in order to prevent **Epic** fro m being manually resolved while some of its Stories are still "**Open**" or "**In Progress".**

Example: Automatically resolve an epic when all its stories are resolved

We are going see how to implement a mechanism in our workflows, in order to auto-transition "**Epic**" to **Resolved** status when all its "**Stories**" become **Resolved** or **Closed**. Let's see in screenshots how to configure the different features.

We add **Block or hide a transition for an issue depending on its issue links** to "**Resolved Issue**" transition in "**Epic's** workflow:

Issue link types:	☐ is blocked by ☐ blocks ☐ is cloned by ☐ clones ☐ is duplicated by ☐ duplicates ☐ has Epic ☑ is Epic of ☐ relates to ☐ relates to Selected issue link types will be allowed. Nevertheless, if you didn't select any issue link type, and checked *"Allow unselected issue link types"*, there won't be applied any filter by issue link type, i.e., every issue link type will be allowed.
Issue types for linked issues:	☐ Bug ☐ Epic ☐ New Feature ☑ Story ☐ Task ☐ Improvement ☐ Sub-task Linked issues of 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.
Statuses for linked issues:	☐ Open ☐ In Progress ☐ Reopened ☑ Resolved ☑ Closed ☐ To Do ☑ Done Linked issues 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.

Resolutions for linked issues:

- ☐ ***UNRESOLVED***, i.e. no resolution value
- ☐ Fixed
- ☐ Won't Fix
- ☐ Duplicate
- ☐ Incomplete
- ☐ Cannot Reproduce
- ☐ Done
- ☐ Won't Do

Linked issues 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 issue resolution will be allowed.

Linked issues must belong to:

- ◉ any project
- ○ current project
- ○ any but current project
- ○ projects introduced:

Introduce a comma separated list of project keys. If you write field codes (format *%{nnnnn}*) they will be replaced with its values.

- ☐ ignore links to issues in other projects

Filtering by field values:

Optional boolean expression that should be satisfied by linked issues. (Syntax Specification)

```
1
```

Leave field empty for no filtering. [Line 1 / Col 1]

Logical connectives: **or**, **and** and **not**. Alternatively you can also use **|**, **&** and **!**.
Comparison operators: **=**, **!=**, **>**, **>=**, **<** and **<=**. Operators **~**, **!~**, **in**, **not in**, **any in** and **none in** can be used with **strings**, **multi-valued fields** and **lists**.
Logical literals: **true** and **false**. Literal **null** is used with "**=**" and "**!=**" to check whether a field is initialized, e.g. ***{00012} != null*** checks whether **Due Date** is initialized.

String Field Code Injector:

Summary - [Text] - %{00000}

Field Code for **Current** Issue | Field Code for **Linked Issues**

Numeric/Date Field Code Injector:

Original estimate (minutes) - [Number] - {00068}

Field Code for **Current** Issue | Field Code for **Linked Issues**

Example 1: boolean condition *{00012} <= ^{00012}* will require that linked issues have *Due Date* equal or later than current issue's *Due Date*.
Example 2: boolean condition *%{00074} ~ ^%{00074} AND ^%{00017} in ["Blocker", "Critical"]* will require that linked issues have *Fixed versions* contained in current issue's *Fixed versions* and *Priority* is *Blocker* or *Critical*.

Minimum required number of issue links:

0

Minimum number of linked issues required to satisfy selected filtering conditions (issue link type, issue type, status, resolution, project and field values).

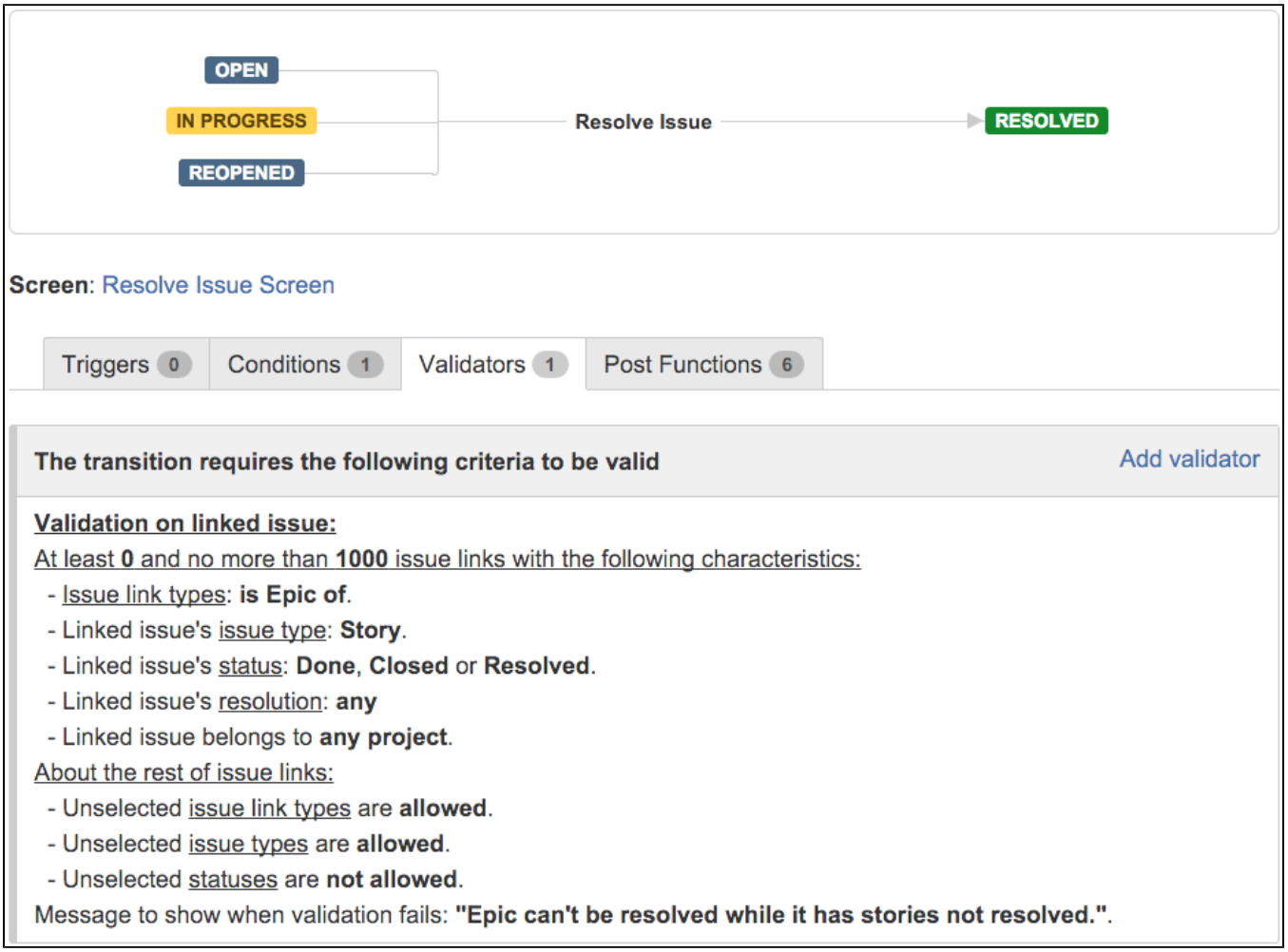
Maximum allowed number of issue links:

1000

Maximum number of linked issues allowed to satisfy selected filtering conditions (issue link type, issue type, status, resolution, project and field values).

Allow unselected issue link types:	☑ Unselected issue link types will be ignored, i.e., they will be allowed regardless of linked issues' issue type, status, resolution, field values and number. Nevertheless, if none of the issue link types are selected, checking this option will make the condition behave as if you had selected every issue link type.
Allow unselected issue types:	☑ Linked issues in unselected issue types will be ignored, i.e., they will be allowed regardless of their status, resolution, field values and number. Nevertheless, if none of the issue types are selected, checking this option will make the condition behave as if you had selected every issue type.
Allow unselected statuses:	☐ Linked issues in unselected statuses will be ignored, i.e., they will be allowed regardless of their resolution, field values and number. Nevertheless, if none of the statuses are selected, checking this option will make the condition behave as if you had selected every status.
Allow unselected resolutions:	☑ Linked issues in unselected resolutions will be ignored, i.e., they will be allowed regardless of their field values and number. Nevertheless, if none of the resolutions are selected, checking this option will make the condition behave as if you had selected every resolution.
Allow unsatisfied condition on field values:	☐ Linked issues not satisfying filter by field values will be ignored, i.e., they will be allowed regardless of their number.
Message to show when validation fails:	Epic can't be resolved while it has stories not resolved. Set translations for installed languages

Once configured, transition "**Resolved Issue**" in **Epic's** workflow will look like this:



We add **Write field on linked issues or sub-tasks** post-function to "**Resolve Issue**" transition in "**Story's** workflow:

Source value that will be written into target field:

Select a source type:

○ **field** in current issue ◉ **parsed text** (**basic** mode) ○ **parsed text** (**advanced** mode) ○ **math** or **date-time** expression

[Line 1 / Col 15]

```
1 Resolve Issue
```

Field codes with format `%{nnnnn}` will be replaced with the corresponding field values. With **Cascading Select** fields use `%{nnnnn.0}` and `%{nnnnn.1}` for referencing **base** level and **child** levels respectively.

Check Syntax

String Field Code Injector:

Summary - [Text] - %{00000}

Field Code for **Current** Issue | Field Code for **Linked Issues / Subtasks**

Numeric/Date-Time Field Code Injector:

Original estimate (minutes) - [Number] - {00068}

Field Code for **Current** Issue | Field Code for **Linked Issues / Subtasks**

Target field that will be set in linked issues or subtasks:

Execute transition (delayed execution) - [Workflow transition]

☐ Don't overwrite target field if it's already set.

Filtering by issue link type:

- ☐ is blocked by
- ☐ blocks
- ☐ is cloned by
- ☐ clones
- ☐ is duplicated by
- ☐ duplicates
- ☑ has Epic
- ☐ is Epic of
- ☐ is caused by
- ☐ causes
- ☐ relates to
- ☐ relates to

Only issues linked to current issue by selected issue link types will be written.

Write also subtasks fulfilling condition on issue type, status and project:

☐

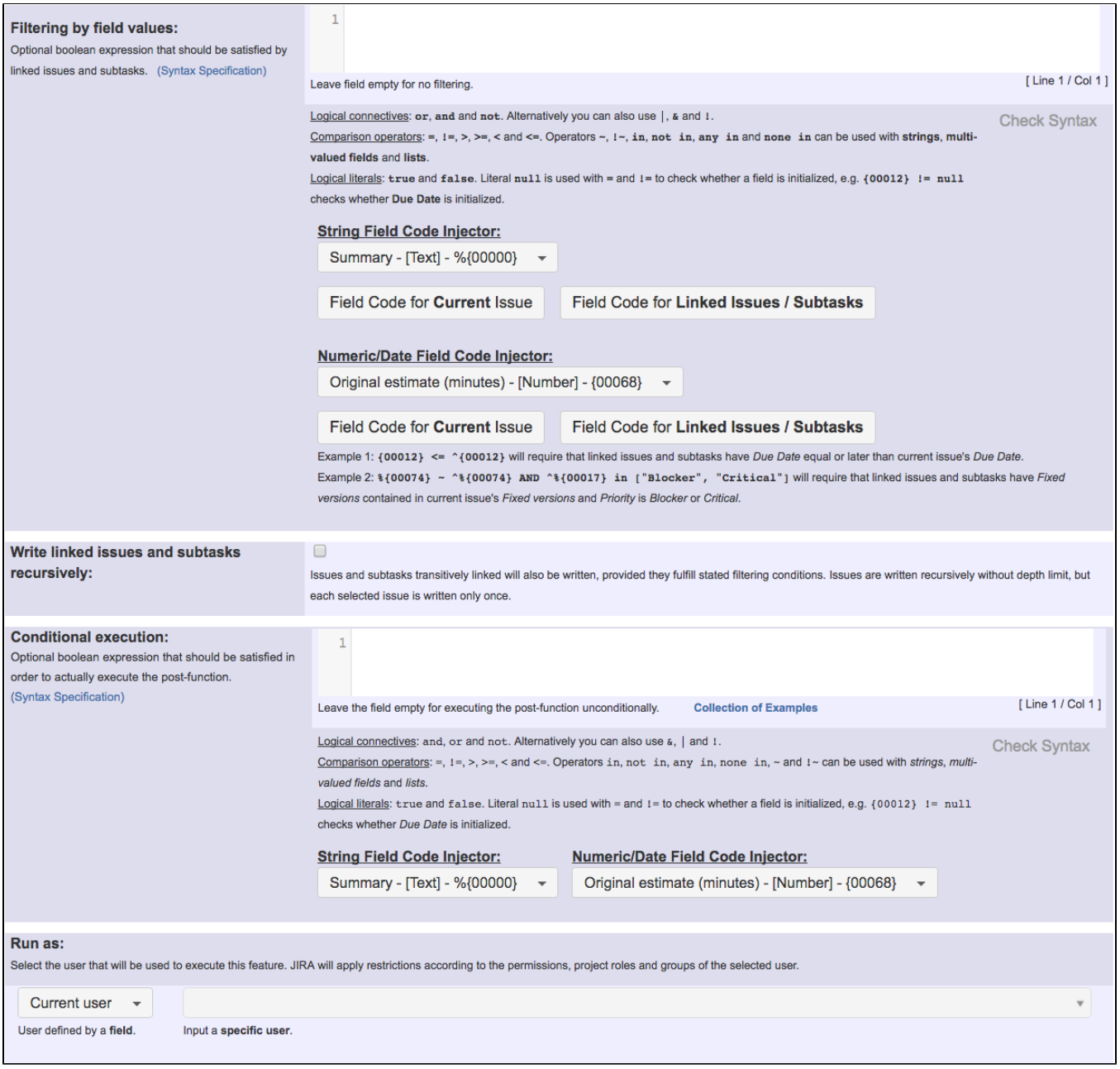
This option only makes sense when current issue itself is not a subtask.

Write also sibling subtasks fulfilling condition on issue type, status and project:

☐

Sibling subtasks are understood as subtasks with the same parent as current issue. This option only makes sense when current issue is itself a subtask.

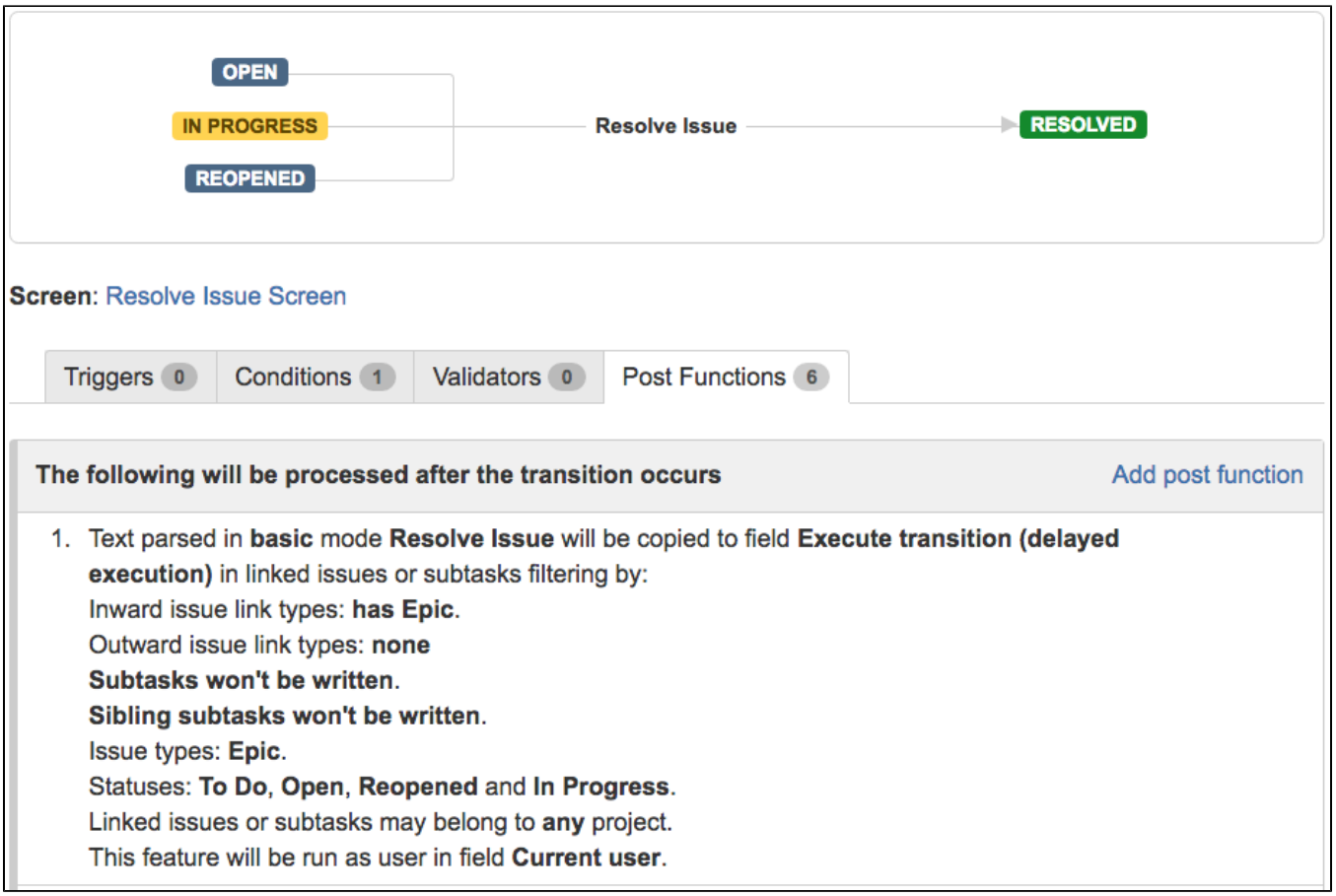
Filtering linked issues or subtasks by issue type:	☑ Epic ☐ Story ☐ Test Plan ☐ Bug ☐ New Feature ☐ Task ☐ Improvement ☐ QA Sub-task ☐ Sub-task Selected issue types will be written, but **if you don't select any, it won't be aplied any filter by issue type**. In that case all the issue types will be written.
Filtering linked issues or subtasks by status:	☑ Open ☑ In Progress ☑ Reopened ☐ Resolved ☐ Closed ☑ To Do ☐ Done ☐ Acceptance ☐ Fail ☐ Pass ☐ Retest ☐ Active ☐ Inactive ☐ Cancelled Selected statuses will be written, but **if you don't select any, it won't be aplied any filter by status**. In that case issues in any status will be written.
Linked issues or subtasks belong to:	◉ any project ○ current project ○ any but current project



Note that: if you are using Jira **7.0 or higher** with versions of **Jira Workflow Toolbox** older than Release Notes for **2.2.8**, you should input the following boolean expression in parameter **Filtering by field values**:

```
count(filterByStatus(filterByIssueType(linkedIssues("is Epic of"), "Story"), "Done, Closed, Resolved")) = count(filterByIssueType(linkedIssues("is Epic of"), "Story")) - 1
```

Once configured, transition "**Resolved Issue**" in **Story's** workflow will look like this:



This same configuration should be added to "**Close Issue**" and "**Done**" transitions in **Story's workflow**, in case direct closing is allowed for stories.

Other examples of that functions

Write field on linked issues or sub-tasks

Page: Add and remove a single or a set of items from multi valued fields
Page: Automatically become watcher of every issue blocking an issue assigned to you
Page: Automatically close resolved sub-tasks when parent issue is closed
Page: Automatically resolve an epic when all its stories are resolved
Page: Compose dynamic text by inserting field values in a text template
Page: Copy "Due date" into a date type custom field in a linked issue if it's greater than current issue's "Due date"
Page: Copy attachments from one issue to another
Page: Create a comment in sub-tasks when parent transitions
Page: Creating a Jira Service Desk internal comment
Page: Creating a Jira Service Desk internal comment on linked issues
Page: Execute transition in epic
Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows
Page: Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
Page: Sum sub-task's "Time Spent" (work logs) and add it to a certain linked issue
Page: Transition sub-tasks when parent is transitioned

Block or hide a transition for an issue depending on its issue links

Page: Automatically resolve an epic when all its stories are resolved

Related Usage Examples

- Block or unblock a transition after an issue rested a specific time in a status
  - example
  - condition
  - validator
  - transition
- Block transition until all sub-tasks are in a specific status category
  - example
  - transition
  - condition
- Validation and condition based on time expressions
  - example
  - condition
  - validator
  - transition
- Validation on sibling sub-tasks depending on issue type and status
  - example
  - validator
  - sub-task
  - transition
- Set a condition in a global transition which only applies in a certain status
  - example
  - condition
  - transition
- Block a transition until all sub-tasks have certains fields populated

Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows

- example
- condition
- validator
- sub-task
- transition

- Block an epic's transition depending on linked issues status and due date
  - example
  - validator
  - issue-links
  - transition
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress" (Transition issues)
  - example
  - post-function
  - transition
- Transition sub-tasks when parent is transitioned
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Transition only a sub-task among several ones
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving sub-tasks to "Open" status when parent issue moves to "In Progress"
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Moving story to "Ready for QA" once all its sub-tasks are in "Ready for QA" status
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Change parent's status depending on sub-task's summary (Transition issues)
  - example
  - post-function
  - transition
- Automatically close resolved sub-tasks when parent issue is closed
  - example
  - post-function
  - sub-task
  - transition
  - outdated
- Change parent's status depending on sub-task's summary
  - example
  - post-function
  - sub-task
  - transition
  - outdated