Restrict sub-task type creation depending on parent issue status

On this page

- Features used to implement the example
- Example: Restrict sub-task type creation depending on parent issue status conditions
- Other variation of the usage example
- Other examples of that function
- Related Usage Examples

Features used to implement the example

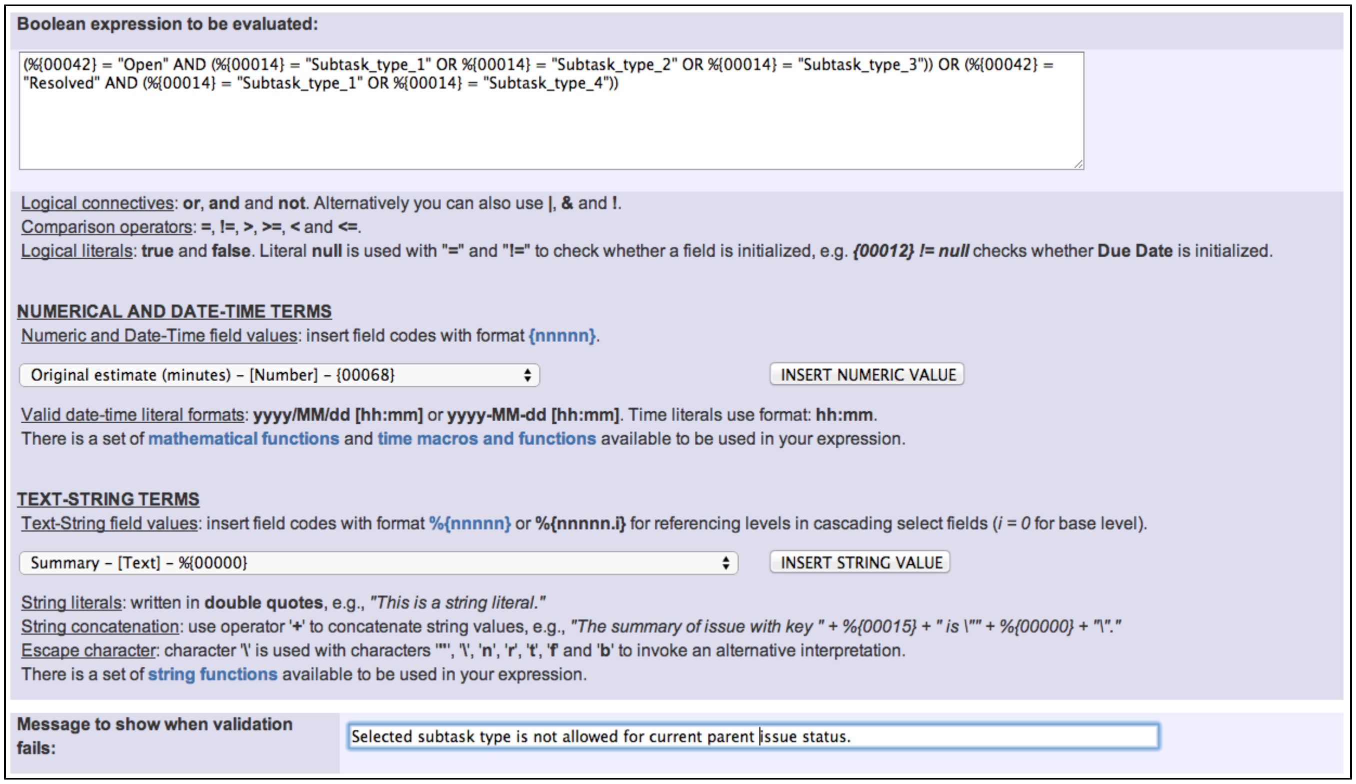
- **Boolean validator with math, date-time or text-string terms**

Example: Restrict sub-task type creation depending on parent issue status conditions

We want to restrict sub-task type creation depending on parent issue status:

1. If parent status is "**Open**" we only allow sub-tasks types "**Sub-task_type_1**", "**Sub-task_type_2**", and "**Sub-task_type_3**".
2. If parent status is "**Resolved**" we only allow sub-tasks types "**Sub-task_type_1**", and "**Sub-task_type_4**".
3. If parent status is different from "**Resolved**" and "**Open**" sub-task creation is not allowed.

We would use the following configuration of **Boolean validator with math, date-time or text-string terms**:



Boolean expression to be evaluated:

```
(%{00042} = "Open" AND (%{00014} = "Subtask_type_1" OR %{00014} = "Subtask_type_2" OR %{00014} = "Subtask_type_3")) OR (%{00042} =
"Resolved" AND (%{00014} = "Subtask_type_1" OR %{00014} = "Subtask_type_4"))
```

Logical connectives: **or**, **and** and **not**. Alternatively you can also use **|**, **&** and **!**.
Comparison operators: **=**, **!=**, **>**, **>=**, **<** and **<=**.
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.

NUMERICAL AND DATE-TIME TERMS
Numeric and Date-Time field values: insert field codes with format **{nnnnn}**.

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

Valid date-time literal formats: **yyyy/MM/dd [hh:mm]** or **yyyy-MM-dd [hh:mm]**. Time literals use format: **hh:mm**.
There is a set of **mathematical functions** and **time macros and functions** available to be used in your expression.

TEXT-STRING TERMS
Text-String field values: insert field codes with format **%{nnnnn}** or **%{nnnnn.i}** for referencing levels in cascading select fields (*i = 0* for base level).

Summary – [Text] – %{00000}

String literals: written in **double quotes**, e.g., *"This is a string literal."*
String concatenation: use operator '**+**' to concatenate string values, e.g., *"The summary of issue with key " + %{00015} + " is \"" + %{00000} + "\"."*
Escape character: character '\' is used with characters '**"**', '****', '**n**', '**r**', '**t**', '**f**' and '**b**' to invoke an alternative interpretation.
There is a set of **string functions** available to be used in your expression.

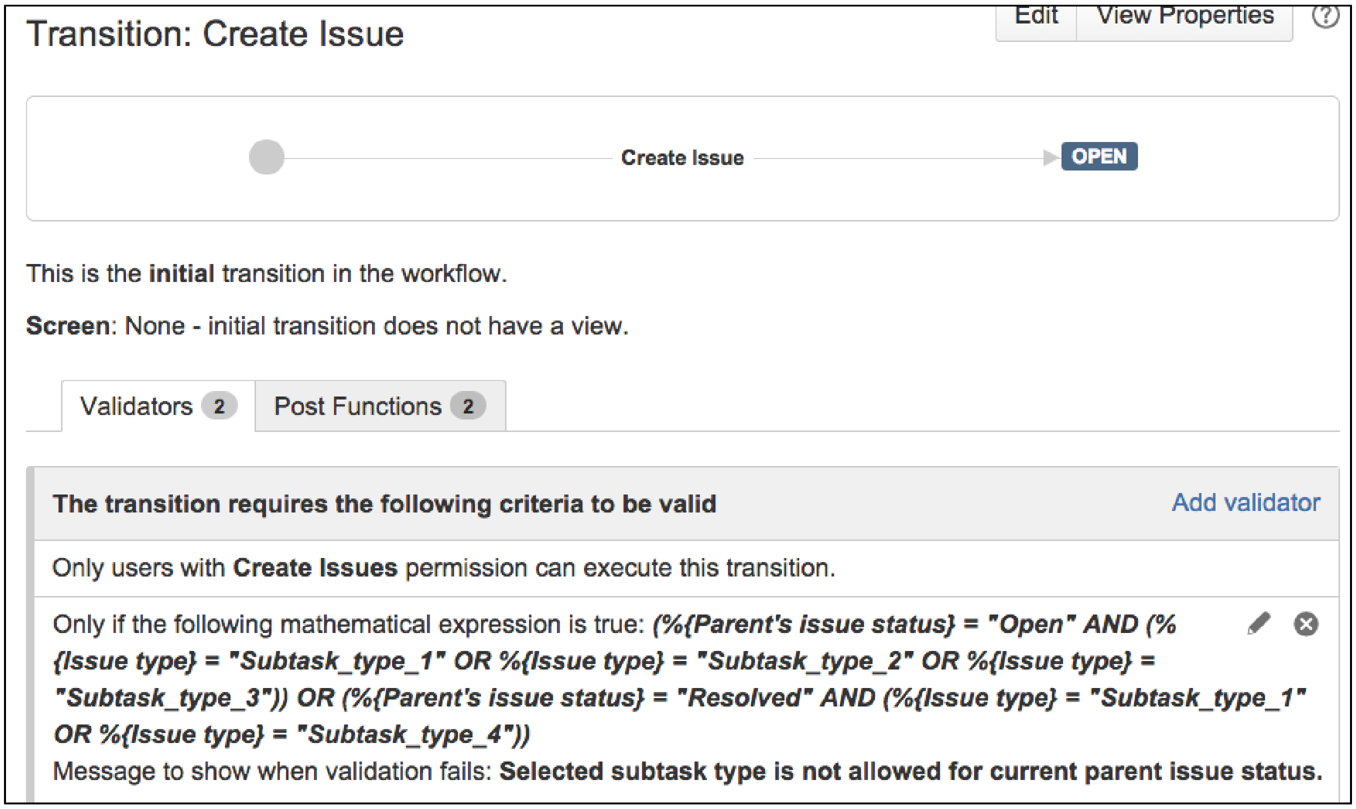
Message to show when validation fails: Selected subtask type is not allowed for current parent issue status.

The boolean expression used: **`(%{00042} = “Open” AND (%{00014} = “Sub-tasks_type_1” OR %{00014} = “Sub-tasks_type_2” OR %{00014} = “Sub-tasks_type_3”)) OR (%{00042} = “Resolved” AND (%{00014} = “Sub-tasks_type_1” OR %{00014} = “Sub-tasks_type_4”))`**

Note that:

- **%{00042}** is field code for **"Parent's issue status"**
- **%{00014}** is field code for **"Issue type"**

Once configured, the transition will look like this:



Transition: Create Issue

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

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

The transition requires the following criteria to be valid

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

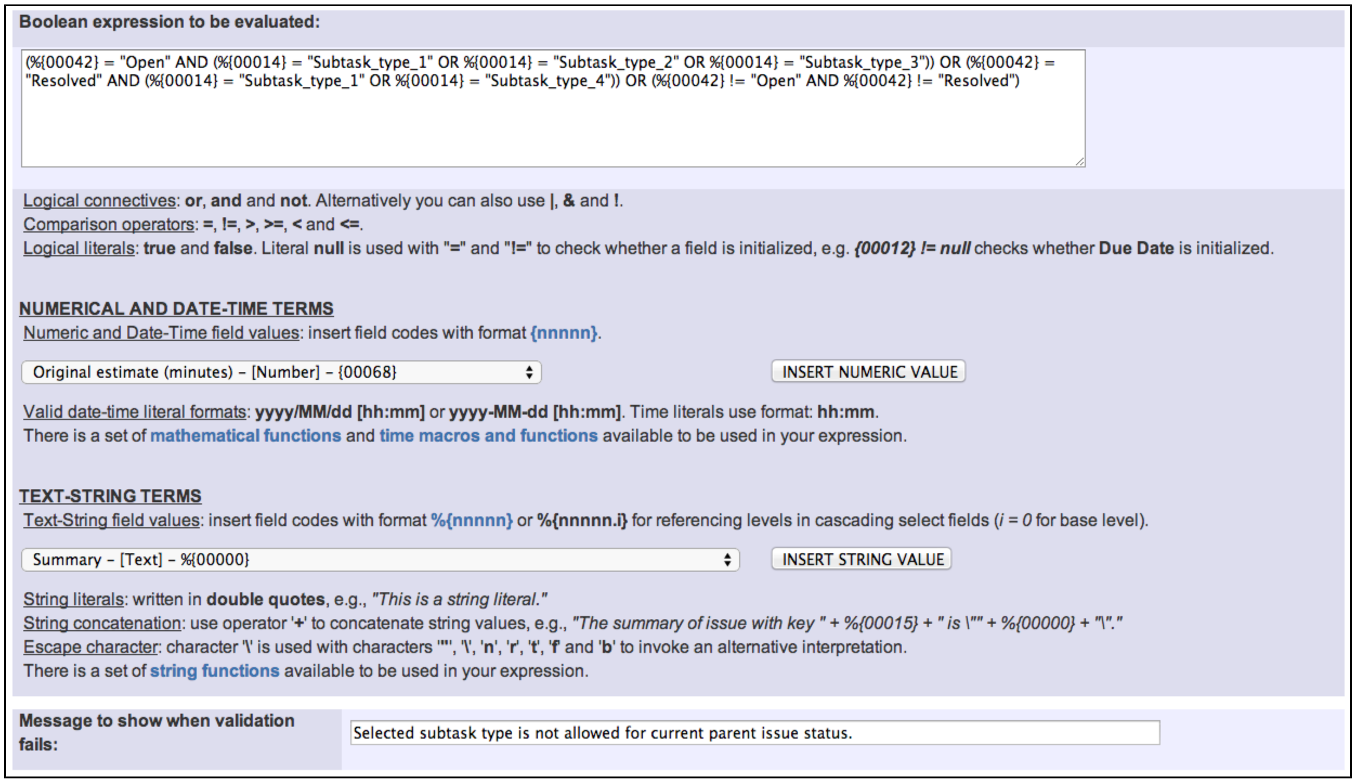
Only if the following mathematical expression is true: ***(%{Parent's issue status} = "Open" AND (%{Issue type} = "Subtask_type_1" OR %{Issue type} = "Subtask_type_2" OR %{Issue type} = "Subtask_type_3")) OR (%{Parent's issue status} = "Resolved" AND (%{Issue type} = "Subtask_type_1" OR %{Issue type} = "Subtask_type_4"))***
Message to show when validation fails: **Selected subtask type is not allowed for current parent issue status.**

Other variation of the usage example

Let us suppose that we replace restriction 3 with:

"If parent status is different from "**Resolved**" and "**Open**" any sub-task type is allowed."

In that case, the following configuration will do the task:


Boolean expression to be evaluated:

```
(%{00042} = "Open" AND (%{00014} = "Subtask_type_1" OR %{00014} = "Subtask_type_2" OR %{00014} = "Subtask_type_3")) OR (%{00042} =
"Resolved" AND (%{00014} = "Subtask_type_1" OR %{00014} = "Subtask_type_4")) OR (%{00042} != "Open" AND %{00042} != "Resolved")
```

Logical connectives: **or**, **and** and **not**. Alternatively you can also use **|**, **&** and **!**.
Comparison operators: **=**, **!=**, **>**, **>=**, **<** and **<=**.
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.

NUMERICAL AND DATE-TIME TERMS
Numeric and Date-Time field values: insert field codes with format **{nnnnn}**.

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

Valid date-time literal formats: **yyyy/MM/dd [hh:mm]** or **yyyy-MM-dd [hh:mm]**. Time literals use format: **hh:mm**.
There is a set of **mathematical functions** and **time macros and functions** available to be used in your expression.

TEXT-STRING TERMS
Text-String field values: insert field codes with format **%{nnnnn}** or **%{nnnnn.i}** for referencing levels in cascading select fields (*i = 0* for base level).

Summary – [Text] – %{00000}

String literals: written in **double quotes**, e.g., *"This is a string literal."*
String concatenation: use operator '**+**' to concatenate string values, e.g., *"The summary of issue with key " + %{00015} + " is \"" + %{00000} + "\"."*
Escape character: character '****' is used with characters '**"**', '****', '**n**', '**r**', '**t**', '**f**' and '**b**' to invoke an alternative interpretation.
There is a set of **string functions** available to be used in your expression.

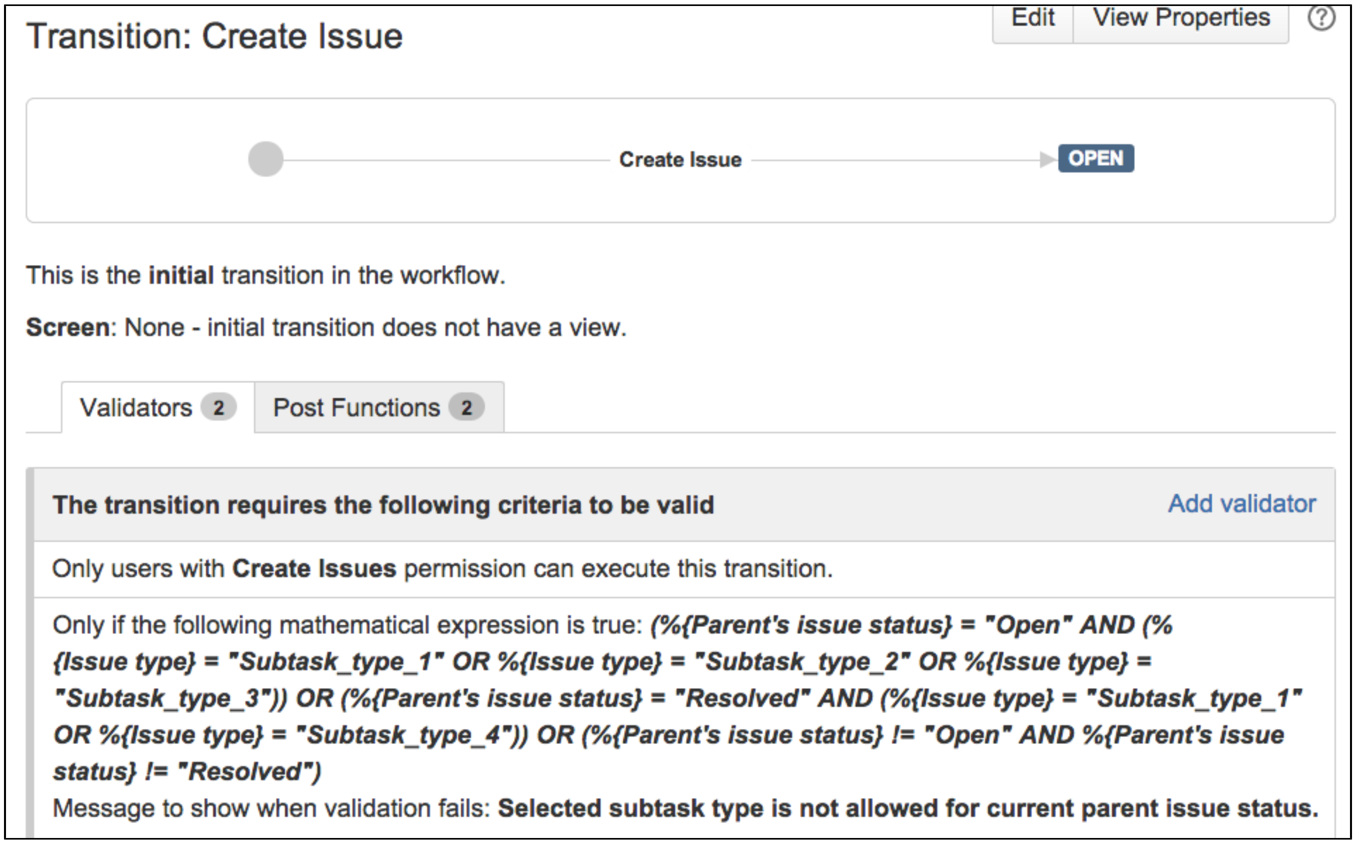
Message to show when validation fails: Selected subtask type is not allowed for current parent issue status.

The boolean expression used: **`(%{00042} = “Open” AND (%{00014} = “Sub-tasks_type_1” OR %{00014} = “Sub-tasks_type_2” OR %{00014} = “Sub-tasks_type_3”)) OR (%{00042} = “Resolved” AND (%{00014} = “Sub-tasks_type_1” OR %{00014} = “Sub-tasks_type_4”)) OR (%{00042} != “Open” AND %{00042} != “Resolved”)`**

Note that:

- **%{00042}** is field code for **"Parent's issue status"**
- **%{00014}** is field code for **"Issue type"**

Once configured, the transition will look like this:



Transition: Create Issue

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

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

The transition requires the following criteria to be valid

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

Only if the following mathematical expression is true: ***(%{Parent's issue status} = "Open" AND (%{Issue type} = "Subtask_type_1" OR %{Issue type} = "Subtask_type_2" OR %{Issue type} = "Subtask_type_3")) OR (%{Parent's issue status} = "Resolved" AND (%{Issue type} = "Subtask_type_1" OR %{Issue type} = "Subtask_type_4")) OR (%{Parent's issue status} != "Open" AND %{Parent's issue status} != "Resolved")***
Message to show when validation fails: **Selected subtask type is not allowed for current parent issue status.**

Other examples of that function

Page: Block a transition until all sub-tasks have certains fields populated
Page: Block an epic's transition depending on linked issues status and due date
Page: Block or hide a transition for an issue depending on its issue links
Page: Block or unblock a transition after an issue rested a specific time in a status
Page: Block transition until all sub-tasks are in a specific status category
Page: Close parent issue when all sub-tasks are closed
Page: Enforce a field (Select List) to be set when another field (Radio Button) has a certain value (works with any kind of field type)
Page: Ensure that all issues linked with a certain issue link type have "Due Date" field set
Page: If field A is populated then, field B must also be populated
Page: Limit issue creation per role and issue type
Page: Limit the number of hours a user can log per day
Page: Limit valid dates for work logs
Page: Make "Time Spent" field required when there is no time logged in the issue
Page: Make a custom field mandatory when priority is "Critical" or "Blocker" and issue type is "Incident"
Page: Make attachment mandatory depending on the value of certain custom field
Page: Make different fields mandatory depending on the value of a Select List custom field
Page: Make linked issues, sub-tasks and JQL selected issues progress through its workflows
Page: Make parent issue progress through its workflow
Page: Prevent issue creation if another issue with same field value already exists
Page: Reject duplicated file names in attachments
Page: Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
Page: Require issue link when resolving as duplicate
Page: Restrict parent issue from closing if it has sub-tasks that were created during a given parent issue status
Page: Restrict sub-task type creation depending on parent issue status

Related Usage Examples

- Validation on sibling sub-tasks depending on issue type and status
  - example
  - validator
  - sub-task
  - transition
- Restrict sub-task type creation depending on parent issue status
  - example
  - validator
  - sub-task
- Require at least one sub-task in status "Resolved" or "Closed" when "Testing required" is selected in Check-Box custom field
  - example
  - validator
  - sub-task
- Restrict sub-task type creation depending on parent issue type
  - example
  - validator
  - sub-task
- Block a transition until all sub-tasks have certains fields populated
  - example
  - condition
  - validator
  - sub-task
  - transition
- Create a dynamic set of sub-tasks based on checkbox selection with unique summaries
  - example
  - post-function
  - custom-field
  - sub-task
- Transition sub-tasks when parent is transitioned
  - example
  - post-function

Page: Restrict sub-task type creation depending on parent issue type
Page: Set a condition in a global transition which only applies in a certain status
Page: Validate a custom field "Story Points" has been given a value in Fibonacci sequence
Page: Validate compatible values selection among dependent custom fields
Page: Validate only issue links created in transition screen
Page: Validate that multi-user picker custom field A does not contain any user in multi-user picker custom field B
Page: Validation and condition based on time expressions
Page: Validation based on the value of a date type project property
Page: Validation on issue attachments
Page: Validation on MIME types of issue attachments
Page: Validation on sibling sub-tasks depending on issue type and status
Page: Validation on the value of a Cascading Select field

    - 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
- Add and remove a single or a set of items from multi valued fields
    - example
    - post-function
    - custom-field
    - issue-links
    - sub-task
- 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
- Moving story to "In Progress" when one of its sub-tasks is moved to "In Progress"
    - example
    - post-function
    - sub-task
    - transition
    - outdated
- Close parent issue when all sub-tasks are closed
    - example
    - condition
    - validator
    - post-function
    - sub-task
    - transition