

Issue#closable? doesn't handle the case of issues with open subtask(s) and being blocked by other open issue(s)

2022/05/09 19:04 - Admin Redmine

ステータス:	New	開始日:	2022/05/09
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	UI_10	予定工数:	0.00時間
対象バージョン:	Candidate for next major release_32	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/33415	status_id:	1
category_id:	10	tracker_id:	1
version_id:	32	plus1:	0
issue_org_id:	33415	affected_version:	
author_id:	1565	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	9		

説明

Subject says it all. Given the current incarnation of @Issue#closable?@, the transition warning for an issue instance that is being blocked while simultaneously having (an) open subtask(s) does not include the 'being blocked' reason (as such including only the 'open subtask(s)' reason).

journals

Here's a preliminary POC fix against current master. Please review and comment.

I'll post an improved and more cleaned-up iteration of this patch later today.

Here's the updated patch that replaces attachment:"0001-POC-fix-for-33415.patch" (which I won't delete for patch evolution review purposes).

The patch is no longer a POC nor is it a final patch yet. There are still some small todo's left.

I ended up re-implementing #31589 in a slightly different manner to let @Issue#closable?@ support the given case of issues with open subtask(s) while simultaneously being blocked by (an)other open issue(s).

Please review and let me know your comments.

Mischa, the scope of this issue is to show in the UI all the reasons for why an issue cannot be closed, right?

Marius BALTEANU wrote:

Mischa, the scope of this issue is to show in the UI all the reasons for why an issue cannot be closed, right?

Yes. At the moment, calling @closable?@ on an issue that meets both the conditions (@descendants.open.any?@ and @blocked?@) returns (somewhat) correctly, but doesn't populate the correct (n)or complete reason.

The cause of this is that @Issue#closable?@ returns from the @descendants.open.any?@ branch, as such not reaching the other @if@-branch. Likewise is support for a transition warning for an issue that meets both the conditions not implemented currently.

Marius BALTEANU wrote:

[...] show in the UI [...]

Not only. The CLI should also return the correct, precise reason.

Marius BALTEANU wrote:

[...] all the reasons [...]

All, except the workflow transition rules. As such can there be four transition warnings in total:

- three transition warnings for @Issue#closable?@:

@descendants.open.any?@

@blocked?@

@descendants.open.any?@ && @blocked?@

- one transition warning for @Issue#reopenable?@:

@ancestors.open(false).any?@

I noticed that I have added more complexity than needed in the previous patches. I'll post an updated patch using a more simplified approach later today.

Here's the updated patch that replaces both attachment:"0001-Fix-33415-by-re-implementing-31589.patch" and attachment:"0001-POC-fix-for-33415.patch" (which I won't delete for patch evolution review purposes).

I don't consider the patch final yet. There are still some small todo's remaining, which I'd like to review and fix-if-needed first.

In this iteration I have removed the unnecessary complexity that I added previously to prevent unnecessarily calling methods multiple times. I simplified the taken approach to handle this, so this complexity wasn't needed any longer.

Some (additional) notes on the patch:

- Code comments added to methods to clearly indicate what functionality they actually provide. A plugin developer looking at what's available for use might expect something other than what's covered by these methods. Maybe we should even shield them for internal use only?
- Replaced short, single condition @if@ structures to use a single-line @if@ modifier.
- For the two conditions: @open_sub_tasks && !open_blocking_issues@ and @open_blocking_issues && !open_sub_tasks@, the negating part is currently optional. If those parts are omitted though, the order of checking for these conditions becomes leading for the result of this method. That might not be a big issue, but it could be broken in the future in case the order in the method is changed.
- Return methods directly from within the @if..elsif..else@ structures now we're using @if..elsif@ structures to check for multiple conditions.

Please review and let me know your comments.

I think this is something small that should be included in 5.0.0 at least.

I'm postponing this because is not such a big issue that we don't provide all reasons from the beginning.

related_issues

relates,Closed,31589,Show warning and the reason when the issue cannot be closed because of open subtasks or blocking open issue(s)

履歴

#1 - 2022/05/10 17:00 - Admin Redmine

- カテゴリをUI_10 にセット
- 対象バージョンを Candidate for next major release_32 にセット