

Redmine - Feature #3517

Assign an issue to person based on the issue status

2009-06-19 22:01 - Felipe Matos Moreira

Status: New	Start date: 2009-06-19
Priority: Normal	Due date:
Assignee:	% Done: 0%
Category: Issues workflow	Estimated time: 0.00 hour
Target version:	
Resolution:	
Description In the same way we can assign an issue to a person based on the category, is it possible to have this ability when we change the status of an issue? I have a need that when a status is changed from Development to Test for example, the issue must be assigned to another person that will conduct the tests. Later, once we marked as Resolved, it should be assigned to another person where it will do the final steps to closing the issue, like moving the code to production environment. Is this feasible? I'm wondering if this could be included as an option in redmine or a plugin. I'm not familiar with Ruby so any help or advices will be very appreciated.	
Related issues:	
Related to Redmine - Feature #521: Add preference to auto-reassign back to au...	New
Related to Redmine - Feature #10139: extend ability to associate settings bas...	New
Related to Redmine - Feature #482: Default assignee on each project	Closed
Has duplicate Redmine - Feature #6708: Change owner when change status	Closed 2010-10-20
Has duplicate Redmine - Feature #10947: Default assignment based on project a...	Closed
Has duplicate Redmine - Feature #19231: Link between "status" and "assigned t...	Closed
Has duplicate Redmine - Feature #13299: Autochange assignee depending on stat...	Closed

History

#1 - 2009-06-30 17:40 - Stéphane Allemand

Hello,

I am interisting with this facility for this case also. Or is it a way to do the same things.

Regards

Stephane

#2 - 2009-07-13 10:37 - J.N. Tang

Good idea! I like it!

#3 - 2011-04-26 18:50 - Pete P

This is a fantastic idea! If implemented, it would allow Redmine to drive a process.

#4 - 2011-04-26 19:17 - Etienne Massip

- Category set to Issues workflow

#5 - 2013-01-12 18:10 - Daniel Felix

Well, this seems to be a good idea. In addition, this could be coupled with the categories.

Example:

1. Ticket with status "new" and category "UI" is created
2. Ticket will be assigned to group "Frontend Developers"
3. Ticket will be set to resolved by a person of the group "Frontend Developers"
4. Status resolve in category "UI" will set the assignment to "Codereview"
5. Ticket will be set to closed by "Codereview" which closes the ticket, or will be set to "Rejected", which will reassign it to the group "Frontend"

Developers"

Example 2:

1. Ticket with status "new" and category "Translation" is created
2. Ticket will be assigned to group "Translators"
3. Ticket will be set to resolved by a person of the group "Translators"
4. Status resolve in category "Translation" will set the assignment to "Translationreview"
5. Ticket will be set to closed by "Translationreview" which closes the ticket, or will be set to "Rejected", which will reassign it to the group "Translators"

Translationreview and Codereview are two different departments in the company and couldn't be in the same group. This way, we could have a complete automatic ticket workflow. What do you think?

#6 - 2013-06-05 20:57 - Ivan Cenov

By now, workflows are defined in transition tables per tracker / per role. This way they are finite state machines. However, these state machines lack of actions on transitions. If such actions were available and was possible to be defined by the user / administrator, above intentions could be fulfilled.

Here is an example list of actions on state transitions:

- change assignee to a previously defined group
- change done percent
- change priority
- change previously defined custom field to a new predefined value
- send e-mail to someone (? - Watcher system is enough?)
- activate external application
- something else...

And as an extension of above, instead of defining single actions, a set (or list) of actions could be defined on state transitions.

I suppose this could be done in a plugin using the hook `:controller_issues_edit_before_save`.

Another idea based on above one idea.

Let's imagine User Interface that presents a state transitions table. Then an user chooses a transition. As a result a dialog activates that shows what will happen on the transition, asking for permission). And if the user confirm changes, they happen and new status is chosen at last.

How this could be activated?

- by a button in issue edit dialog: 'Make a transition' or 'Change status'
- by a menu item in the context menu

#7 - 2013-07-23 23:04 - Christopher Harwood

I'd be interested in broader options for automating transitions between statuses. JIRA provides a good conceptual model for this kind of functionality at <https://confluence.atlassian.com/display/JIRA/Configuring+Workflow#ConfiguringWorkflow-Stepsandtransitions>.

The short version is that a workflow defines both steps (statuses) and transitions (status changes). In JIRA, the definition for a transition includes the target status (what the new issue status will be) as well as one or more of the following:

1. A screen, which provides all (and only) the fields related to the status change. Redmine can nearly replicate this behavior by defining all unrelated fields as "Read Only" for the target status, although setting these in the current UI is a real pain.
2. Conditions, which is just who can change an issue from status X to status Y. Redmine already does this.
3. Input Validators, which Redmine also provides already.
4. Post Functions, or automated changes to the issue that occur at the time of transition. These can include changing the assignee to a specified value, but they can include changes to other issue attributes as well.

If I knew enough Ruby, I would approach the problem in the following way:

Revise the UI for workflows around the concept of transitions; most of this replaces the "Read Only/Required" matrix of field settings in workflow administration. An administrator creates a Transition, and then adds Properties to the Transition. After clicking "Add Transition Property" the user is presented with three types of Properties:

- (1) Field: Fields the user can change when changing the status; these appear in the Update panel
- (2) Required Field: Fields the user *must change* when changing the status, which appear as required fields in the Update panel, and
- (3) Automatic Update: Fields that change automatically; for these the admin defines the new value for the field.

Back in the standard/original workflow UI, there is a space beneath each checkbox and a new column to the right with a list of Transitions. By dragging a Transition over a checkbox and releasing it, the box is checked (if it wasn't already) and the name of the transition appears below the check. (If we want simple but clunky, we could also include a drop-down menu beneath each checkbox.) That enables the selected User Role to perform the selected change from one state to another, but the change follows the rules defined by the assigned Transition. Changes without assigned Transitions, where the admin just checked the box, occur as normal.

Thoughts?

#8 - 2014-03-05 21:00 - Yamuna Vee

+1 - Does anyone know if there's a plugin that implements this feature?

#9 - 2014-11-12 14:29 - Cédric HATEM

Hello,

Any news about this really usefull feature ?

Cédric

#10 - 2014-12-18 16:02 - Mikael Blom

+1 Great Feature

#11 - 2014-12-18 19:34 - markus schulte

Yamuna Vee wrote:

+1 - Does anyone know if there's a plugin that implements this feature?

you may want give http://www.redmine.org/plugins/redmine_luxury_buttons a spin. /m

#12 - 2015-01-31 11:54 - Hans Kaiser

+1 for the request to implement the missing feature

markus schulte wrote:

Yamuna Vee wrote:

+1 - Does anyone know if there's a plugin that implements this feature?

you may want give http://www.redmine.org/plugins/redmine_luxury_buttons a spin. /m

The plugin looks great. Is anyone using it already?

It provides all the features, which redmine is lacking right now about ensuring status processes.

Any plans from the redmine dev lead to incorporate the plugin in default redmine?

#13 - 2015-04-14 16:20 - Juan José Rosales

This plugins works!!!!

https://github.com/jjrosalesuci/redmine_auto_assigned

#14 - 2015-05-26 12:56 - David Molina

Can you improve the documentation, please?

#15 - 2015-07-01 10:29 - Toshi MARUYAMA

- Related to Feature #482: Default assignee on each project added

#16 - 2015-07-01 12:10 - Toshi MARUYAMA

- Has duplicate Feature #19231: Link between "status" and "assigned to" fields added

#17 - 2015-07-01 12:15 - Toshi MARUYAMA

- Has duplicate Feature #13299: Autochange assignee depending on status change added

#18 - 2016-06-01 23:05 - Alfredo Bonilla

+1

#19 - 2016-08-27 07:20 - M Hasan

The link of https://github.com/jjrosalesuci/redmine_auto_assigned is not working, page not found

#20 - 2016-08-29 12:34 - JH Lee

/M Hasan

You're right.

The link of https://github.com/jjrosalesuci/redmine_auto_assigned is not working, page not found.

Does anyone know an active link of downloading the plugin of this feature?

#21 - 2017-04-21 04:39 - dl rm

+1 Does anyone know an active link of downloading the plugin of this feature?

#22 - 2017-04-23 17:38 - Yuuki NARA

This plug-in can be used for automatic setting of assignee in charge of workflow.

http://www.redmine.org/plugins/status_button

The original corresponds to 2.3, but github-forked version supports 3.2.

I hope you find it useful.

#23 - 2017-04-26 04:14 - dl rm

thanks, I'm researching...

Yuuki NARA wrote:

This plug-in can be used for automatic setting of assignee in charge of workflow.

http://www.redmine.org/plugins/status_button

The original corresponds to 2.3, but github-forked version supports 3.2.

I hope you find it useful.

#24 - 2017-10-13 13:54 - Gerhard .

+1, would be interested in this feature too.

Has anyone experiences with the plugin Status Button in conjunction with V3.x?

Is there any other plugin available which is not in alpha stadium since 4 years?

regards

gerhard