Redmine - Feature #21930

Change priority depending on the due date

2016-02-02 22:15 - Dominik Bochenski

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Issues	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Hello,

is this possible to automatically change priority of task depending on the due date?

For example:

due date = 10.01.2016

today = 05.01.2016
priority = Normal

today = 08.01.2016
priority = High

today = 08.01.2016
priority = Urgent

History

#1 - 2016-02-02 22:17 - Dominik Bochenski

Sorry,

today = 10.01.2016
priority = Urgent

#2 - 2016-03-30 15:03 - davide davide

Hi, I'm interesting about this features.. I have the 2.3.0 version on my server and this could be a important plugin if exists.

At the moment I can extract priority/start date/due date and I change the priority with a sql script... but this isn't the better way for me!!

Thanks

#3 - 2016-04-29 08:24 - Toshi MARUYAMA

- Category changed from Project settings to Issues

#4 - 2018-06-14 15:08 - Maik Lindner

...would be better with a "RoR layer" but you can create a daily mysql event with:

```
CREATE EVENT prio_issues_with_due_date
ON SCHEDULE EVERY 1 DAY
STARTS '2018-06-14 05:00:00'
DO
update issues i
join
issue_statuses s ON s.id = i.status_id
set
priority_id =
    case
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.due_date,'%Y-%m-%d')) =-2 and priority_id <
= 4 then 4
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.due_date,'%Y-%m-%d')) =-1 and priority_id <
```

```
= 5 then 5
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.due_date,'%Y-%m-%d')) = 0 and priority_id <
= 6 then 6
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.due_date,'%Y-%m-%d')) > 0 and priority_id <
= 7 then 7
        else priority_id
    end
where
s.is_closed = 0
and
i.due_date is not null
;

ALTER EVENT prio_issues_without_due_date
ON SCHEDULE EVERY 1 DAY
STARTS '2018-06-14 05:00:00'
DO
update issues i
join
issue_statuses s ON s.id = i.status_id
set 
priority_id = 
    case 
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.created_on,'%Y-%m-%d')) = 1 and priority_id
 <= 4 then 4
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.created_on,'%Y-%m-%d')) = 2 and priority_id
 <= 5 then 5
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.created_on,'%Y-%m-%d')) = 3 and priority_id
 <= 6 then 6
        when datediff(date_format(now(),'%Y-%m-%d'), date_format(i.created_on,'%Y-%m-%d')) > 3 and priority_id
 <= 7 then 7
        else priority_id
    end
where
s.is_closed = 0
and
i.due_date is null
;
```

Remember that there won't be any history.

#5 - 2018-06-14 17:07 - Bernhard Rohloff

Changing the priorities directly in the database feels like an evil hack to me. I would rather create a "Redmine Bot" account with elevated privileges and use that account to change the priority of the issues by a script over the REST Api or by a rake task (see reminder emails as example). If the issues priority gets elevated by the bot, it is logged in the issue journal and so it's a documented change by the system. Additionally the bot can add a note about the reason of the change and involved users are getting notified by email.