Redmine - Feature #661

Allow uni-directional relations between tasks

2008-02-13 23:12 - Jeremy L

Status:	Closed		**Start date:**	2008-02-13
Priority:	Normal		**Due date:**	
Assignee:	Jean-Philippe Lang		**% Done:**	0%
Category:	Issues		**Estimated time:**	0.00 hour
Target version:				
Resolution:	Fixed			

Description

This relates to [#638](#638), issue relations between projects:

I have one redmine project for people outside the company to submit issues and features.
Then, I have another project that is for internal use by engineer(s).

The two levels allow me to have a development plan that determines which public issues become internal issues, and lets me assign and prioritize them appropriately. The mapping is not one-to-one. For some small tasks, the internal and external issues may be the same. However, Larger external requests might be broken into smaller internal tasks. Some external requests might be ignored. Some internal tasks (like server maintenance) might be added without an external request.

The public should never be able to assign work directly to engineers, and only certain engineers should be able to reply publicly to customers.

The ugly solution is copy & paste issues between projects. But this maintains no linkage, and requires duplication of data. Relations are a better way, but they need an additional feature:

Relations would need an option to be 'one-way' or unidirectional. E.G. A task on Project A refers to a task on project B, but B does not make any mention of the link to A. So an internal worker can see a reference to the external request, but people outside the company do not see the links to internal issues.

Just hiding the reciprocal link, when requested, might be enough to implement this feature.

Going one step further, a person with the correct Role & permissions (such as a manger), could see the link from both the internal and external project (it might appear different to indicate it is otherwise invisible). This way, it becomes easy for them to check if a related task has been resolved.

History

#1 - 2008-02-13 23:50 - Jeremy L

A related solution might be this: Add a Permission called "view issue relations", in addition to the existing Permission called "Manage issue relations". Then, certain users, such as those on the external project, do not have permission to see any relations.

This is not as powerful, but has some of the same effect.

#2 - 2008-02-14 22:35 - Jean-Philippe Lang

- Assignee set to Jean-Philippe Lang

IMHO, a simple (and maybe cleaner) solution would be to hide the relation if the user is not allowed to view the related issue.

Eg. user X can see the project A with issue #1 but he can't see project B.
Issue #1 is related to issue #2 of project B.
Since X shoudn't be able to see #2, he shouldn't be able to see the relation between #1 and #2.

Currently, relations are displayed in any case. That could be considered as a bug since user X is not allowed to view #2.
What do you think ?

#3 - 2008-02-14 22:39 - Jean-Philippe Lang

One more thing: in my solution, if user Y is allowed to view both #1 and #2 he will see the relation on #2 and on #1 as well, which can be convenient for him.
With the solution you propose, he wouldn't see the relation when looking at #1 although he is allowed to see #2.

Don't know if it's clear...

#4 - 2009-02-26 06:10 - Brad Rushworth

I didn't like the solution proposed by Jeremy L but Jean-Philippe Lang's solution seems elegant, simple and managable.

Any progress on this implementation? I think a lot of people would find this useful (me for one).

#5 - 2009-03-19 10:33 - Rob van den Bogaard

I'm also in favor of the solution proposed by Jean-Philippe Lang. We introduced Redmine at our office, and have several interacting sub-projects (e.g. "Community site", "Backoffice app"), so we would really like being able to relate issues of different projects.

#6 - 2010-05-16 16:27 - Felix Schäfer

- Status changed from New to Closed

- Resolution set to Fixed

AFAIK this has been implemented, i.e. one only sees relations to issues one has the permission to view.