Redmine - Feature #1266

Feature: Allow setting multiple target-milestones

2008-05-20 19:01 - Sepp _

Status:	New		**Start date:**	2008-05-20
Priority:	High		**Due date:**	
Assignee:			**% Done:**	0%
Category:	Roadmap		**Estimated time:**	0.00 hour
Target version:				
Resolution:				

Description

this is needed when we discover a bug and it needs to be fixed in the stable branch and in the development-branch....
It is difficult and not very much the desired resultto copy each issue for each milestone...

I think, the issue could internally be seen as "several issues"...., like #1234m1 for milestone 1??
When it gets fixed for milestone [#1](#), i'd like to set milestone one as fixed. If i search all tickets for milestone 2,
this issue should still show up and should not be seen as fixed. Maybe it would be nice to see it 95% fixed, so i know, that you just
have tu merge the code and test it....

what do you think?

Related issues:

Related to Redmine - Feature #1675: Add 'affected version' as a standard field	**Closed**	**2008-07-23**
Related to Redmine - Patch #219: Issues fixed in multiple versions	**New**	
Related to Redmine - Patch #5510: Enable Mutliple Versions Per Issue	**Resolved**	**2010-05-12**
Has duplicate Redmine - Feature #5272: Allow multiple target versions	**Closed**	**2010-04-08**

History

#1 - 2008-05-20 20:43 - Carl Nygard

Considering how simple it is to copy the issue and reassign the milestone, I can't see the benefit of multiple milestones. From a user perspective there might be one extra mouse click.

Since you specifically state that internally it "would be seen as several issues" I think it's much clearer and cleaner to make it **in fact** multiple issues using the copy mechanism.

Setting a relation between the two issues means the original bug information is just a click away. It's quick and easy, and fixing a bug for a release vs. development is actually two different operations, and so really **should** be two different issues.

#2 - 2008-05-20 21:52 - Sepp _

the dissadvantages when copying are:

- you can't see when you are looking a an issue, in which versions it is fixed.
  --> loss of overview
- you need to know several issueIDs when checking in the fix for an issue and you want to include hat in the committext
  --> again, loss of overview, especially when looking a your commit-messages. you have to check all issue#s
- when you have more than 2 target milestones, it is more work to copy an issue and set it to the different milestones!
  --> and even more clicks
- if you have to change the primary message because the reported error is different than the actual error, you have to
  write this in every copied issue....
  --> more work, more klicks

--> generally, it would be nice to see, if every fix of SP2 made it into the next stable-version... this would be
possible in this case...

#3 - 2008-05-20 23:46 - Anonymous

To start with, this is a duplicate of issue [#284](#). I even wrote a patch to implement this behaviour (see [#219](#)); however, there is no way that this patch would cleanly apply anymore - things have changed far too much since then. There is also a [forum post](#) that is asking these same questions, and finding issue duplication to be a "long and annoying procedure".

> Considering how simple it is to copy the issue and reassign the milestone, I can't see the benefit of multiple milestones. From a user perspective there might be one extra mouse click.

Duplication of this kind of data is just plain wrong. Updating one ticket would now mean updating the other 4 tickets (for example), but how does the user know which other issues to update unless a whole bunch of issue relations were created in the first place? Even then, is the user really going to remember? In our organisation, issues get handed back and forth with extra information being added all the time. It's just not feasible to have multiple issues for the same "issue".

There is most definitely a case for being able to specify multiple targets for an issue, but in my case, we do not require that issues get marked as fixed in one target or another. If we have an issue that needs fixing in 3 branches, we use the Refs commit keyword for the first two, then Fixes for the last one. We do not need an automatic progress update or commit messages such as #1234m3. Our target names are not really suitable for putting into a commit message like that last example - they are long, contain white space, etc...

All we need is to say an issue is part of multiple targets so that they show up in each target's roadmap, etc. You can see a brief overview of the workflow that my company utilises in [#284](#). Currently, writing release notes for a version is far too difficult - if an issue is fixed in version 2.1 and 1.1, the issue is currently marked as an issue in r1.1 with a note for fixing also in 2.1. Listing the issues for 2.1 does not show the issues that were fixed yet marked for 1.1.

Another place in our organisation where Redmine is not quite there yet, is how our QA team mark that an issue has been fixed in one version, but remains untested in another version. The only way I can think this would be workable though is actually having an issue per target to facilitate this need for having a different status in each target. I don't like the idea at all, but I don't know how to answer this problem at the moment either :)

#4 - 2008-08-16 00:44 - Thomas Pihl

We are creating "link-issues" per branch/version (almost like sub-issues) to keep track.

We do NOT want to copy issue since comments/updates in one branch can be critical information for another branch's update (usually some weeks/months later). The bad side of link issues is the loss of good overview on roadmap/issue summary pages. Cut/Paste of subject works as long as no branch-team changes/clarifies it. Metrics/summary are partly lost (we cannot keep link-issues as a specific tracker since we use different workflows in different branches/versions (different rules when changing a live version compared to a untested/future release)). We can tell metris per version/branch but not unique issues for the whole system.

Perfect solution would be some special lightweight link-issues with a subset of attributes (maybe their own workflow, status, timetracking and perhaps branch-local-comments) but with all other fields/comments linked back to main issue. Perhaps some way to close top-issue when all sub-issues are closed.

/T

#5 - 2008-11-11 10:36 - Jean-Philippe Lang

- Target version deleted (0.8)

#6 - 2009-01-14 01:20 - Reach Everywhere

+1

#7 - 2009-01-14 09:41 - Maxim Krušina

I'm strongly again: Milestones are often used also for non-spftware purpose. If you assign more than one milestone to single ticket, time tracking will be in big trouble, because spent time of single ticket will be calculated two (or more) times to different versions resulting in wring time and price calculations which is critical.

For example, we're using milestones for different blocks of project and we're billing milestones to our clients separately.

#8 - 2009-01-14 11:53 - Anonymous

Maxim Krušina wrote:

> I'm strongly again: Milestones are often used also for non-software purpose. If you assign more than one milestone to single ticket, time tracking will be in big trouble, because spent time of single ticket will be calculated two (or more) times to different versions resulting in wring time and price calculations which is critical.
>
> For example, we're using milestones for different blocks of project and we're billing milestones to our clients separately.

And that is fine - you would simply continue to not use multiple milestones for your work. The situation in other companies **requires** multiple milestones/versions/branches/releases per issue in order to track changes. I would suggest a setting for showing a single or multiple selection control for the issue version.

#9 - 2009-02-05 11:30 - Thomas Pihl

Isnt this another issue that could be solved with subissues (se #443), each issue having their own status, target version and everything else useful. Timekeeping isnt the only problem with multiple milestones. How to know what if some step has been done for a specific branch(future version)?

/T

#10 - 2009-02-06 10:29 - Sepp _

I don't need subtask as I don't need Timetracking for all different Milestones..
When I solve a task for one branch, I just have to merge the code to the other branch.... so timetracking is not always needed...
I just need to document if I merged the change in both branches or if I forgot....
So, adding a second target-milestone the same way as adding a new watcher would be the best solution for me. Internally,
the 2nd. Milestone could be a 1:1 copy of the main ticket with only 2 seperate fields: It needs its own Milestone And its own Solution..

#11 - 2009-11-12 00:47 - Alex Last

is there a way to vote for issues in this redmine?
We really need this "multiple target" feature. we have several branches and need to enforce the process somehow that bugs need to be fixed in all of
them before marking as "resolved/closed".

#12 - 2010-01-25 18:45 - Rocco Stanzione

Been a while, but I'd like to +1 this. I may implement it myself locally.

#13 - 2011-07-23 11:22 - Terence Mill

In general multiselect switch feature should be possible for all fields, also custom fields of type list or enumeration.

#14 - 2013-04-16 12:08 - Toshi MARUYAMA

- Category changed from Issues to Roadmap

#15 - 2013-06-11 20:11 - Joel SCHAAL

Ok so I added "my vote" to [#5272](), which was marked as duplicate of this one.
But my remarks are still the same :

> It has been entered more than ~~2~~ 5 years (!!) ago and it is still new.
> Is this a rejected issue ?
> Does this issue depend on other work ?
> Can we have some update on what will be decided for this task ?

Thank you.

Edit : The discussion [Versions vs Milestones]() could be related to this issue.

#16 - 2013-10-31 09:33 - Ketki Vahalia

+1
In our case, we track issues by customer versions and also by server patch versions. Making a copy and remembering the relations is needless
overhead for developers. There should be a clean way to accomplish the need of associating an issue with multiple versions. Thanks!

#17 - 2013-11-01 14:06 - Michaël de Groot

+1.

We have several live versions running of a project. 1 issue unfortunately may be fixed in several versions.

#18 - 2014-02-07 21:30 - Alcides Silveira Costa

+1

#19 - 2014-06-09 15:48 - Natanael Copa

We (alpine linux) are currently creating multiple issues to track security bugs in all our 4 stable branches. It means that for each security issue we
create normally 5 issues (one tracker issue and one for each branch). With 5 issues (a normal week) this becomes 25 issues and much time goes to
simply create and close those tickets.

I would **love** beeing able to only create one ticket for each security issue (which after all are same issue) and simply mark which git branches needs
to be fixed instead of making 5 copies of same thing. It would make our lives much easier.

Thanks!

#20 - 2014-07-16 11:18 - Alice Etchegaray

+1

#21 - 2014-07-30 17:45 - David Fouche

+1

#22 - 2014-08-19 20:42 - Domingo Galdos

+1

#23 - 2014-11-27 11:22 - Toni Smillie

I really need this too - please prioritize it for upcoming version

#24 - 2014-12-17 13:51 - Eduardo Lima Fabricio

+1 .. I need this too !

#25 - 2015-05-01 21:26 - Todd Van Allen

This is a very vital feature for organizations with multiple releases to manage.

#26 - 2015-09-22 16:33 - Ricard F

+1000!!

Vital feature!

#27 - 2015-09-23 13:08 - budo kaiman

+1

#28 - 2015-09-23 17:32 - Todd Van Allen

I'm guessing the reason that this feature (which is pretty key) is not getting much traction as the work-around is creating your own custom field (e.g. 'Targeted for Versions') and using that. But this creates two such fields in the UI which is confusing. A Key field like this one should be multi-select.

#29 - 2015-09-23 17:37 - Sepp _

Todd Van Allen wrote:

> I'm guessing the reason that this feature (which is pretty key) is not getting much traction as the work-around is creating your own custom field (e.g. 'Targeted for Versions') and using that.

Not only. Every 'Targeted for Versions' hat its own status, so it might be fixed in version 3.0, but is still open in version 2.0.

#30 - 2015-09-23 18:00 - Todd Van Allen

The hack that we use is the addition of two other fields; 'Affects Versions' and 'Fixed in Versions'. This tracks the versions that the application is broken in and which versions it's been fixed in. The delta between these two fields provides the 'Known Issues' for a particular release. It's an ugly hack with a lot of overhead, but it's functional at this point. Until, of course, they add this feature to the application.

#31 - 2016-07-27 22:29 - Mark Wintch

As of today in this environment I am not able to perform this function. We have also purchased the RedmineCRM Agile plugin which does not allow multiple target versions to be selected. I will also follow up with that group, however this seems a good place to request this feature be implemented in Redmine as with the 3.3.0 release I still am unable to do it. Are any others getting this done? I do not want to do a work around, however please explain it if you can as we need some way of implementing this feature to the greatest extent possible.

Thank you.

Environment:
Redmine version   3.3.0.stable
Ruby version   2.1.8-p440 (2015-12-16) [x64-mingw32]
Rails version   4.2.6
Environment   production
Database adapter   SQLServer
SCM:
Filesystem
Redmine plugins:
redmine_agile   1.4.1

#32 - 2017-09-05 12:53 - David LECLERCQ

I think too, it's a vital feature.

When can be realised this feature ?

Thanks

#33 - 2018-07-06 08:30 - Edgars Batna

+100
This should be somewhere on the Roadmap, as managing simultaneous maintenance and development teams is currently a pain.

It is not so much about creating tickets per version as the absolutely retarded usability that comes with it. We need a solution to stop copying information around and have a usable view about ONE ISSUE that has sub-issues per version, which does not increase our blood pressure.

Should be some sort of inheritance of properties or fields and a sensible view for such "target version subtickets". I'm sure there is a technical solution to this.

Plus, if one looks at the Roadmap, there's so many useless, low-level, unimportant technical things getting changed, yet the elephant remains.

#34 - 2020-08-17 04:05 - Qiang Gao

It seems this feature hasn't been done yet, it's 13 years and I think it's harder and harder to add this feature since it would change the design of Redmine, what a pity

#35 - 2020-08-26 11:17 - Marco Shima

We are also missing this feature. Actually there is no feature we're missing more at the moment.

#36 - 2021-06-18 13:38 - Mark Tomm

+1

This is a must have for non-weekend projects which span multiple OS and architectures.

#37 - 2024-02-21 06:02 - Marco Shima

Mark Tomm wrote in #note-36:

> +1
>
> This is a must have for non-weekend projects which span multiple OS and architectures.

Exactly my opinion.