Redmine - Feature #6366

Due date on an issue should follow the associated release due date if it exists

2010-09-11 20:50 - Tony Jacobs

Status:	New	**Start date:**	2010-09-11
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

When an issue does not have an end date assigned, but it does have a version assigned, and the version does have a date assigned, then the end date of the associated bug should be implicitly assumed to be the release date.

I think it's a misfeature to want the upper bound of an existing end date to be clamped to the release date, so I'm explicitly not asking for that here. Just when the end date is undefined / null.

Observed in Redmine 1.0.1 against MySQL 5.0.91

Related issues:

Related to Redmine - Patch #251: Patch for Feature Request #9785 (Default iss...	**New**	
Related to Redmine - Feature #7626: version due date and issue due date inter...	**New**	**2011-02-15**
Related to Redmine - Feature #5451: Make Start Date/Due Date settable to vers...	**New**	

History

#1 - 2010-09-12 06:36 - Burt Culver

try http://www.redmine.org/wiki/1/PluginIssueDueDate

#2 - 2010-09-13 04:08 - Tony Jacobs

This plugin looks like it will work for what I'm trying to do right now, but it's not the 'right way' to solve the problem. The 'right way', I believe would be to leave the due date field on the issue itself as undefined. The subtle distinction becomes apparent when moving an issue from version XX (due yyyymmdd) to version FUTURE (not due ever)

#3 - 2010-09-13 22:58 - Terence Mill

I share Tony's opinion.

The question shall be:

Does a ticket with a "due date > assigned projects due date" every would make sense?
If a ticket has no due date, and gets set automatically the projects due date, one can never find such tickets which has ever explicit scheduled.

#4 - 2010-09-14 04:20 - Tony Jacobs

Terence Hersbach:
Consider this case:

Feature *foo* is required for a major deliverable on August 1.

Release alpha is due on May 1, and *foo* is included in that release's requirements because it makes sense/risk reduction/whatever. It is a candidate to slip to Release beta on June 1, or to Release gamma on July 1.

Here, we can easily see that since *foo*'s due date is well beyond release alpha, we can slip it to a follow-on release.

#5 - 2010-09-14 11:33 - Terence Mill

Well, i think you mixing two kind of versioning. Product (marketing) release versions (where features are officially available for customer) and software release, what means the feature is implemented in code.
Another thing you have to imagine is, that every software version cycles from requirements>planning>implementing>testing (acceptance, integration)> production. In Enterprise busines you mostly have no alpha or beta released to production, so the release on may1 or any before August1, will go into tesing environments

version May 0.8 (alpha)
due date: May 1

release environment: customer acceptance test
- Feature "foo" implemented

version 0.9 (beta)
due date: June 1
release environment: Customer system integration test
- BUF FIX for "foo" resolved (tickets from acceptance test)

version 1.0 RC
due date: August 1
release environment: Customer production
- BUF FIX for "foo" resolved (tickets from integration test)