

ステータス:	New	開始日:	2008/09/25
優先度:	通常	期日:	
担当者:		進捗率:	0%
カテゴリ:	Issues planning_40	予定工数:	0.00時間
対象バージョン:	Unplanned backlogs_23	作業時間:	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/1953	status_id:	1
category_id:	40	tracker_id:	2
version_id:	23	plus1:	8
issue_org_id:	1953	affected_version:	
author_id:	2212	closed_on:	
assigned_to_id:	0	affected_version_id:	
comments:	28		

説明

We have been using FogBugz (<http://www.fogcreek.com/FogBugz/>) for some time and I think it's a really good (commercial) solution. A few days ago a friend told me about Redmine, I was trying it and I have to say I am in love with it (geek? Nooo...)

We just need one specific functionality we have in FogBugz and it's very important for us. I'll try to explain it:

- Each version must have a new field: 'Start Date'.
- Each user (developer) configures on his own account the days and the hours he spends working on the project. We will need some new tables/fields here.
- When an issue is assigned to the user, he estimates the time in hours. We can use the existing 'Estimated time' field.
- Based on this data (start date, user time and user estimations) we can calculate (on the fly) the version 'Due Date'.
- We can calculate the due date for each user and for the whole version. And may be for the whole project too (Is it too much?).
- If the issue is reassigned, if the user changes the estimation or if somebody changes the start date, then the due date is updated automatically.

This is probably a simple feature but we think it's very useful for the managers or the marketing people involved in the project.

There is another related functionality FogBugz implements and it's really great. It saves logs with the spent time on each issue by the user and it uses these logs to create some kind of 'profile' of the user with his estimations accuracy. This allows us to create a probability sheet with the due date, eg: the due date will be at '10/20/08' with 25% chance and '10/28/08' with 50%, and '11/15/08' with 75%, and '12/31/08' with 99% (100% doesn't exist ever). To create this, we need to know when the user starts or stops to work on each issue. May be we can use the 'start/stop' button asked by Anna Labinskaya or any other way (<http://www.redmine.org/boards/1/topics/show/1027>).

This is not so simple but we can build it when we finish the first feature.

Please, let me know your questions or concerns. I can add some mockups and/or screenshots if I was not enough clear.

Thank you!

journals

I'm all in favour of this too.

This is the one reason that I nearly chose FogBugz over Redmine (we may still switch - nothing is ever set in concrete!), and it's a really important one at that.

I vote for this as well.

I am using Redmine until I can cost-justify Fogbugz. I also prefer the method Fogbugz uses of the developer inputting the number of hours remaining instead of percent completion.

I created this feature request 2 month ago and I have no response from your side. We really think it's a good and very useful feature and we can contribute to the project developing if you agree.

Btw... Thomas, I assigned it to you because you answered my question on the forum (<http://www.redmine.org/boards/1/topics/show/2447>)

+1 - I also request theses features!

I think there is a lot of people who would like this feature but nobody from redmine answered or commented the issue. May be we can build it as a plugin but I think the best way is to add it to the main structure. Let me know what you think.

Mauricio Miranda wrote:

May be we can build it as a plugin but I think the best way is to add it to the main structure. Let me know what you think.

Building a plugin would be best. Once the plugin is ready, we can see about adding it to the core.

+1 for this feature from me as well. Is anyone currently working on this? Just wondering if this is actually under development in the mainline or as a plugin, or if it's stalled. I might be willing to contribute some time to trying to implement this if it's not currently going anywhere.

Maybe the `[[PluginSchedules|Schedules]]` plugin fills your request partially?

I banged out an attempt at this feature and submitted a pull request from edavis10 on github. We'll see if it makes it in, as it's my first attempt at contributing to Redmine. I'll post an update here if the functionality does make it in, and if it doesn't maybe I can figure out how to make a plugin that would add the functionality that way.

David Davis:

Could you post a link and/or patch here? I'm the only one that gets pull requests and I don't want to roadblock this issue if I'm busy.

Posting patch; this is based on redmine
09e47a3b6325c109c63a70dc47a9a8e32e762b44 as head.

Can this issue be implemented into one of next releases?
I think it's very important, let it be optional, but it'll be a big plus to Redmine.

Sorry for my English :)

Is this implemented or not. We are looking for it desperately .

+1

+1

Especially we like the estimation accurancy profile thing. This will create a retroperspective on every version cycle and leaning curve for better estimations and make them more expert. In fact it should be possible to have them anonymous per feature/version or per personalized.

Great idea!

+1

I also totally require this feature as well. But I would like an additional or different feature.

It would work the following:

For a Milestone (or Version in Redmine) I define Issues. I would specify the time estimate, an assignee and a unique priority number (e.g. added as a custom field).

Now, Redmine should be able to compute the start and end dates for each issue in the version, based on priorities (higher priority number first), issue dependencies (relations, predecessors first) and assignees (issues can be done in parallel by different people).

I would also be willing contribute, since I'm a developer myself. (Unfortunately I'm not into Ruby, but c++. Maybe I could help to write the algorithm, when having some of your expert-ruby-assistance. :))

I really require this! :) Or was it already added meanwhile? Answers and comments are highly appreciated!!

+1

This sounds amazing! Would love to see it implemented.

+1

+1

Definitely need this!! Currently we have to do our planning on MS Project, then add all the same tasks to Redmine using the dates into MS Project gave us.

Each time we need to move the start date of a task (and believe me, that happens a LOT, mainly when we work on our in-house projects on our free times) we have to manually change all the dates. Think of that when a project has 100+ tasks, it takes at least 2hs to update all the tasks...

at least to be able to choose between due date or duration

All my +1s to this. Frankly, this is huge, and I'm a bit irritated that I've spent the last couple of days trying to get Redmine to do this. Using duration as an alternative to explicit due dates is a basic feature of every project management tool I've ever used. It didn't even occur to me that you simply wouldn't be able to do it in Redmine. Lacking this basically prevents it from being usable by me, so I really hope this gets some attention soon. :(

Here's my solution. It assumes 8 hour workdays and that Saturdays and Sundays are skipped as non-work days. It should be easy to adapt if your work schedule deviates from this.

Add the 'Custom Workflow' plug-in:

<http://www.redmine.org/plugins/custom-workflows>

The Custom Workflow plug-in is extremely powerful and you can use it in many ways to customize Redmine to a pretty high level. It basically executes custom code before and after saving an issue. Create a new custom workflow as follows.

Description:

When Start Date or Estimated Hours changes, updates the issue Due Date. Saturdays and Sundays are skipped and it's assumed that there are 8 Estimated Hours per work day.

Code block (left side):

```
if leaf? && self.start_date && self.estimated_hours
  if start_date_changed? || estimated_hours_changed?
```

```

self.due_date = self.start_date + (self.estimated_hours+7)/8
non_work_days = 0
(self.start_date..self.due_date).each do |date|
  if date.wday == 0 || date.wday == 6
    non_work_days += 1
  end
end
self.due_date += non_work_days + 2*(non_work_days/7)
end
end

```

----- Brian Lindahl wrote: > Here's my solution. It assumes 8 hour workdays and that Saturdays and Sundays are skipped as non-work days. It should be easy to adapt if your work schedule deviates from this. > > Add the 'Custom Workflow' plug-in: > FANTASTIC!!! ----- Brian Lindahl wrote: > [...] Thanks for Brian. It's a big help. But in some situations the due day will fall into weekends by your code. I made a revision to this. And in my redmine, the due date is the last day for the job, not the last+1 (Didn't use ruby before. If something is wrong, plz correct me.)

```

if leaf? && self.start_date && self.estimated_hours
  if start_date_changed? || estimated_hours_changed?
    work_days = ((self.estimated_hours + 7.9)/8).to_i

    if work_days <= 1
      self.due_date = self.start_date
    else # >= 2
      work_days -= 1 # check from the next day.

      cur_date = self.start_date + 1 # move to next day
      while work_days > 0 do
        if cur_date.wday == 0 || cur_date.wday == 6 # if next day is weekend
          cur_date += 1
        else
          work_days -= 1
          self.due_date = cur_date
          if work_days > 0
            cur_date += 1 # move to next day
          end
        end
      end
    end
  end
end
end
end
end

```

There is also a plugin that will assign the due date based on versions and durations:

https://github.com/railsformers/redmine_issue_due_date

related_issues

relates,New,2687,different estimated time units hours<>days per tracker type

履歴

#1 - 2022/05/10 17:27 - Admin Redmine

- カテゴリ を Issues planning_40 にセット

- 対象バージョン を Unplanned backlogs_23 にセット