redmineorg-copy202205 - Vote #67508

Auto schedule issues and Estimated end date for versions

2022/05/09 14:47 - Admin Redmine

ステータス:	New	**開始日:**	2010/06/29
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	Issues_2	**予定工数:**	0.00時間
対象バージョン:		**作業時間:**	0.00時間
Redmineorg_URL:	https://www.redmine.org/issues/5774	**status_id:**	1
category_id:	2	**tracker_id:**	2
version_id:	0	**plus1:**	0
issue_org_id:	5774	**affected_version:**	
author_id:	16654	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	4		

説明

Calculating an "Estimated end date" for a version/milestone allows you to both track delays during the process and use it for the initial planning of due dates for the versions.

Making a real estimation of the end date requires automatic scheduling of the issues assigned to a programmer. This has another benefit that the Gantt diagram and Calender reflects what is actually on right now, so you always see an updated working plan. (it is quite annoying with the current solution that you have to re-assign dates for all issues every time something is not following the plan)

Here is how I think it should be done:
1) Each programmer specify how many hours he can work every week day (e.g. 8 hours mon-friday).
Preferably this layout of hours is put into the calender for this programmer so he can modify the planned hours for specific days (i.e. if he takes a day off, is on vacation or can work extra on specific days).

2) The start/due dates for the issues assigned to a programmer, is automatically re-assigned so each day is filled up, based on his hours set above. This happens every time the Status or %Done is changed for an issue. Only issues that are not yet i progress will have their start date re-calculated.
The workload (number of hours left) for Issues in progress is calculated from how many percent is completed.
Issues are assigned in the order they should be completed (first comming milestone/version first, and within a version: order by priority)
To do all this precisely it is required that Issues includes Time with their Start date. See issue #5458.

3) Under each milestone/version you can see both the due date (still manually assigned) and the expected date it will be completed.
Below is a list of all programmers that have issues assigned with this version. For each programmer you can see his expected end date. This allows the manager to quickly overview if one programmer is delaying the version - either because he is overloaded or is working to slow...
The "Expected end date" for this version is simply the end date for the programmer with the last expected end date.

4) The Gantt chart should reflects the updated dates and times for the issues, so the programmer can use it as a working plan and the manager can get a real time overview of the progress and planned tasks.

This feature references issue #1953
I consider this to be a very important feature. Please consider it for version 1.1.

journals

I vote for this feature.
As an issue reporter, after creating and assigning a new issue, I have no idea what are the assignee's current tasks are, but I would like to see an estimation when does my issue is going to be started and finished.
Redmine's current "Start" field must be filled manually and recalculated manually on every change of an issue with higher priority. I imagine this as a new "*Estimated start*" redmine field that is re-calculated automatically on every change of any issue with the same assignee.

For example, today is 23-DEC-2010, the current time is 14:00

A developer Doe with 8 hours/day load, starts working at 9 am, is having 3 issues assigned to him:
issue # 1, priority Urgent, status:In progess, estimated time:10 hours, 30%done, "*Estimated start*":nil (since status is "In progress")
issue # 2, priority High, status:New, estimated time:1 hour, "*Estimated start*": 24-DEC-2010 15:00 (is calculated automatically since he has to spend 7 more hours for issue # 1)
issue # 3, priority Normal, status:New, estimated time: 7 hours"*Estimated start*": 24-DEC-2010 16:00
Etc.

If you create a new issue with priority High and assign it to the user, "*Estimated start*" for all the issues with lower priorities assigned to this user will be re-calculated automatically.

Bump, this is a feature I'd love to see for both myself and the advancement of the software's scope.

The only thing that ms project has over Redmine in my book is auto task scheduling based on resource (programmer) working hours and availability in other projects.

I really think this would bolster Redmine significantly in the team management area and really speed up entry and re-organising of issues.

To allow for flexibility and backwards compatibility with Redmine users you could allow tasks to be in auto or manually scheduled mode (as ms project does quite well).

I'd see it as two parts, a resource time management plugin and the auto scheduling logic plugin and agree with Hans' proposed implementation.

Obviously this feature is most useful to a project manager but would also be useful to programmers as a work queue and for time efficiency.

I understand that this is quite a large feature and would increase the scope and user audience of the Redmine project but I really think for the better.

thanks for the great effort thus far, Redmine is fantastic!

I strongly vote for this issue, too. It would make Redmine a killer project management software (while it currently is "just" a superb project management software).

Actually, I'm even consider digging into Redmine development and implement it myself.
Since the rule-sets are relatively straight-forward, it should not be THAT hard, I think.
The easiest solution would be re-using and updating the original due field and introduce, as Michael Walton explained, a "manual" and "automatic" planning mode for tasks. In this case, the GANTT logic does not need to be touched at all.

I did some work on this at the weekend. Of course, I first tried to dig into the Redmine source code, but quickly realized that I don't understand Ruby at all. So, as a proof of concept, I tried to solve this problem on the MySQL database just using database tools (namely procedures and triggers). I have to say that the solution I came up with until now is not that bad. I introduced a new table `user_worktimes' to store they hours a user works per weekday and calculated an issue's due date from the start date and the estimated time. Related tasks (followers, blocked by) are automatically schedule after the due date of the latest predecessor.
This works pretty well, but still has some limitations and does not work automatically. I still have to call the stored procedure auto_plan_project(), which, of course, is not desireable.
I plan on digging into Ruby and the Redmine source code now, but it seems like a lot of efforts.
Anyways, just for inspiration, I attached the SQL scripts that I used for my proof-of-concept, db-based solution to the problem.

Execute auto_schedule.sql, then auto_plan_project_draft.sql (MySQL only).

履歴

#1 - 2022/05/10 17:22 - Admin Redmine

- カテゴリ を Issues_2 にセット