redmineorg-copy202205 - Vote #62943

Update % to be programatic not arbitrary

2022/05/09 12:58 - Admin Redmine

:	New		:	2008/03/29
:			:	
:			:	0%
:	Issues_2		:	0.00
:			:	0.00
Redmineorg_URL:	https://www.redmine.org/issues/952		status_id:	1
category_id:	2		tracker_id:	2
version_id:	0		plus1:	1
issue_org_id:	952		affected_version:	
author_id:	679		closed_on:	
assigned_to_id:	0		affected_version_id:	
comments:	25			

Right now % is a completely arbitrary value entered by the issue assignee. Fact is, more often than not, it is totally off and pretty much a useless metric. I would prefer to see % be a programmatic value derived from estimated time vs logged time and have someone update the estimate if it looks like they have further to go. That will help keep estimated time (and hence budget) more accurate.

journals

There are two ways to accomplish this:

- Having a configuration option on the project (or the tracker) named "compute done ratio" and letting Redmine update the value when appropriate. In that case the value should not be editable by the user.

Or, simpler, but with a short delay:

* Doing this outside of Redmine using an SQL-Update statement that is called regularly.

I would like to add that it would be great if tickets could optionally aggregate the percentages from their child-tickets automatically.

We often work with "container-tickets" or "meta-tickets" to group related tasks.

For example, for an imaginary project with the goal to "create a website" we might create the following structure:

Ticket #O1 (assigned to project manager): Create a Website
Ticket #O2 (child of #O1, assigned to Graphician): Create logo images for $website
Ticket #O3 (child of #O1, assigned to Coder): Program logic for $website

Now the graphician and coder will start working and (ideally) update the percentage on their individual tickets regularly. It would be very helpful for the project manager if the parent-ticket (#O1) could reflect the aggregate percentage of all subtickets.

Thus, when ticket #O2 is 50% done and ticket #O3 is 90% done then the parent ticket should read: 70% done.

For the implementation I'd imagine a simple checkbox next to the percentage bar in the ticket-view. This checkbox should only be visible when a ticket has any children and when enabled it would cause the percentage to be computed as described.

The tricky part and what really turns this into a killer-feature (for us) is

that this kind of "percentage inheritance" should ofcourse work for a ticket-tree of unlimited depth. The percentages of the children of children would aggregate into the child and the percentages of the children would aggregate into the parent ticket, etc.

That way a project manager could easily "drill down" into the ticket-tree to identify bottlenecks and take countermeasures.

This is a little bit like the "Milestones"-view but we don't want to create Milestones for every little thing. Instead our Milestones consist of a number of "subprojects" that evolve around such ticket-trees that I just described.

Ah, forgot a tiny detail that's necessary to make the "drill down" truly effective:

The percentage of child tickets should be displayed next to the ticketname in the "related tickets"-section of the ticket-view.

That way you can immediately see which of the child-tickets have a low percentage without actually opening each one individually.

Sorry for stupid question, but how you are doning "ticket nesting"? I don't see any implementation for this, except for ticket releation..

Yes, by relation; "Child of" and "Parent of".

Just bumping this ticket so it doesn't get lost.
I like the ideas, are they being considered?

Sorry for stupid question, :-) but what if estimated_hours is less than spent_time? What to draw? 140%?

I can not speak for Thomas but my idea would be to make *two* bar-graphics: One for "Percentage done" and one for "Time used".

I would leave the "Percentage done" value manual (with the proposed ticket-inheritance) and apply the automatic that Thomas proposed only to the
"Time used"-graphic. And yes, if the ticket goes overtime then I would draw the graph 100% full, make it red and have the value next to it display the percentage (e.g. 140%).

My reasons:

- "Percentage done" can not be automatically determined. A human has to go in and decide how much is done or left to do.
- "Time used" on the other hand can be computed as Thomas proposed.
- The two values are independent and both are important. "Percentage done" is important to monitor progress and my proposed inheritance would help a lot with that. "Time used" is important for monitoring the timeline, keeping track of developer assignments and budgets.
- I think everybody will agree that "time spent" is not directly related to "percentage done", especially in software-projects.

PS: I'm Moe Mataz. Had to create a new account because I could not login any more and password reset did not work.

We ended up building a plug-in to handle this with a slightly different approach. The program allows the admin to set a % done value associated with each status. It makes it more general, but less arbitrary and the user no longer needs to fuss with % done. This is super useful when building reporting based upon % done. The key for it to be reasonably accurate is to make sure that tasks are not gigantic multi day tasks but smaller bit sized one - never longer than .5 day's time. If anyone is interested in the patch, I could ask eric to release it is we haven't done that yet.

ex:

proposed: 0%
new: 0%

in progress: 50%
feedback: 70%
pending signoff: 90%
complete: 100%

etc etc

+1 Our organization is interested in the patch. Entering in percent done seems redundant if you are already updated the status. I like this fix.

shane pearlman wrote:

> We ended up building a plug-in to handle this with a slightly different approach.

It'll be open-sourced? If so...

When/where we can get it? (-;

I'm sorry folks, It looks like I offered prematurely. We are a bit too slammed at the moment to go through the qa process and sync it up with the current open sourced release. Perhaps in a few months.

peace

I'm assigning this issue to myself to help remind me about this plugin. I'll see if I can Open Source it soon.

Moe Mataz wrote:

> Yes, by relation; "Child of" and "Parent of".

Hi, sorry for the stupid question, but I can't find the "Child of" and "Parent of" type of relations

All I have is (I'm translatinf from spanish) : related with, copy of, blocks, before.

And I could't find where to edit the types of relations

am I missing something???

sebastián scarano wrote:

> Moe Mataz wrote:
>
> > Yes, by relation; "Child of" and "Parent of".
>
> Hi, sorry for the stupid question, but I can't find the "Child of" and "Parent of" type of relations
>
> All I have is (I'm translatinf from spanish) : related with, copy of, blocks, before.
>
> And I could't find where to edit the types of relations
>
> am I missing something???

No, you're not missing anything, the relationship types in redmine are just a little bit, ehm, wierd. For example "Blocks" and "Precedes" seem to be basically redundant, but that's a different story...

Back on topic: For a parent/child relationship you basically make the child "block" the parent. In an ideal world the parent could then not be closed before all children are closed and the parent would automatically compute the "Percentage done" from the children's values. Furthermore tickets with such a relationship would show as a tree (with indentation) under the "issues"-view.

Ofcourse we don't live in an ideal world and redmine does nothing of that as of today. ;-)

---

mmm

I'm beginning to think it won't be so easy to add new relations type

I've just seen it's hardcoded in

http://www.redmine.org/repositories/entry/redmine/branches/0.7-stable/app/models/issue_relation.rb

how come you've got "child of" and "parent of" relations???

sebastián scarano wrote:

> mmm
>
> I'm beginning to think it won't be so easy to add new relations type
>
> I've just seen it's hardcoded in
>
> http://www.redmine.org/repositories/entry/redmine/branches/0.7-stable/app/models/issue_relation.rb
>
> how come you've got "child of" and "parent of" relations???

oops, my apologies to Jim Jones, I posted before reading his answers...

I've just created a patch for this on #4274. It makes an issue's %done to be managed by it's Issue Status.

using issue statuses to calculate %%done (task #4274) looks very (VERY) weird.
my feature has 10 subtasks, which are estimated to take 10 hours each. so my feature should take 100 hours.

why would I see 50% done right after I just started working on a feature or on any of the subtasks? what does this 50% number tell you? Maybe I'm missing something, but it looks absolutely useless and WORSE - misleading.

I agree with Alexey. It looks weird to calculate % done by using issue statuses.

Any news about using log time and estimate time?

A previous poster had a good idea. Have a separate "Time Used" field, which could be more than 100% Another option would be to have the "%Done" operate as "Time Used" by a config setting.

Referring to my previous post, "Time Used (%)" = log_time / estimate_time * 100

---

related_issues

relates,Closed,4274,Use Issue status for percent done
relates,New,12762,Add option to calculate done ratio with the data from time tracking
duplicates,New,2561,Correlate workflow with Issue complete percentage
duplicates,Closed,4116,Automatic status propagation

#1 - 2022/05/10 17:29 - Admin Redmine

-              *Issues_2*