

Redmine - Defect #21449

Automatic done ratio calculation in issue tree is wrong when parent has its own estimated time

2015-12-07 20:27 - Sebastian Paluch

Status:	New	Start date:	
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Issues	Estimated time:	0.00 hour
Target version:		Affected version:	3.2.0
Resolution:			
Description Due to the change in r14272 estimated_hours are now independent for parent and subtasks. But, it seems that automatic calculation of done ratio is still based on estimated time of children, ignoring value provided for the parent. e.g. <ul style="list-style-type: none">• A (2h)<ul style="list-style-type: none">◦ B (2h)<ul style="list-style-type: none">▪ C (2h) 50% done This results with 50% done ratio on all tasks up to A, what, from estimated time point of view, is wrong - only 1 of 6 hours is "done". I do not see a fix for this when automatic done ratio calculation is enabled as there is no way to say that some work on parent has been also done. Anyway, latest changes leads to inconsistency			
Related issues: Related to Redmine - Defect #16092: Parent/subtask: calculation of estimated ... <div>Closed</div>			

History

#1 - 2016-01-06 04:55 - Toshi MARUYAMA

- Related to Defect #16092: Parent/subtask: calculation of estimated hours added

#2 - 2020-05-25 22:47 - Sebastian Paluch

This issue still exists in Redmine 4.1.0 and this is big issue. Don't know why none complains about this.

#3 - 2020-08-27 20:13 - Sebastian Paluch

It seems that the equation in recalculate_attributes_for has to be changed to include the estimate time of the issue to:

```
total_estimated = p.estimated_hours.to_f || 0.0
children = p.children.to_a
if children.any?
  if p.closed?
    p.done_ratio = 100
  else
    done = children.map {|c|
      estimated = c.total_estimated_hours.to_f
      total_estimated += estimated
      ratio = c.closed? ? 100 : (c.done_ratio || 0)
      estimated * ratio / 100.0
    }.sum
    progress = ((total_estimated - done) / total_estimated) * 100
    p.done_ratio = 100 - progress.floor
  end
  # ancestors will be recursively updated
  p.save(:validate => false)
end
```

The problem with this is that this will correctly update "% Done" of all parents (A) recursively but it will not update "% Done" of currently edited issue (B) if the issue is also a parent (for C).

I have tried moving this recalculation to before_save callback, so that it can modify the issue (B) and this works correctly, "% Done" of current issue (B) and all parents (A) is updated.

This creates though small inconsistency in journals. Normally updates to derived "% Done" are not save into journal. But in the case, when issues (B) is edited and it has derived "% Done" (because it is parent for C), the new "% Done" value gets saved into journal. This is not OK as the "% Done" may be changed at any time if any subtask (C) is modified - since this operation is not be saved in journal, now the issue (B) shows different value of "% Done" then visible in last journal entry. So, confusing. And I don't know how to solve this problem.