# Redmine - Feature #35657

## Spent time ratio column and filter

2021-08-01 16:07 - Go MAEDA

| | | | |
|---|---|---|---|
| **Status:** | New | **Start date:** | |
| **Priority:** | Normal | **Due date:** | |
| **Assignee:** | Marius BĂLTEANU | **% Done:** | 0% |
| **Category:** | Issues | **Estimated time:** | 0.00 hour |
| **Target version:** | Candidate for next major release | | |
| **Resolution:** | | | |

**Description**

I suggest adding a new column and a new filter to the issues list.

The name of the column and the filter is "Spent time ratio". "Spent time ratio" column displays the percentage of a spent time divided by an estimated time. The column is useful to see the sign of overspending of man-hours and find issues that took more time than estimated.

Easy Redmine already has the same column and filter.

spent-time-ratio.png

### History

**#1 - 2021-08-01 16:26 - Marius BĂLTEANU**

Nice and useful idea, indeed! Maybe Easy Redmine can contribute with this feature.

**#2 - 2021-08-13 19:57 - Mischa The Evil**

*- File 2021-08-13_19-25-33.png added*

*- File 2021-08-13_19-26-15.png added*

*- File 2021-08-13_19-26-38.png added*

*- File 0001-Add-a-basic-and-incomplete-POC-implementation-of-tot.patch added*

This indeed looks like a useful addition for users who are doing some more serious planning within Redmine, so last weekend I started hacking together a naive and basic implementation of such a feature.
I quickly came to the realization that this is effectively an alternative implementation of the feature that is requested in all the issues mentioned under DoneRatio_issues_overview#Requests-and-patches-for-adding-3rd-option-Use-the-logged-and-estimated-time, but then implemented as separate column(s) instead of re-using the "% Done" column and without the limitation of the value at 100%. So, if this feature is really something we want to add to the Redmine core we would have to consider it in light of the potential implications if (one of) those are (considered to be be) implemented in the core too.

I'll attach a patch for demonstration and testing purposes only. This patch adds a basic and incomplete POC-implementation of (total) spent_time_ratio columns. It does not yet provide:

- Column sorting functionality;
- Filters for the added columns;
- Rendering of (total) spent_time_ratio values in CSV, PDF & API output-formats;
- I18n;
- Test coverage;

Given the current (Issue)Query sorting implementation, I don't see a way to easily add the first two missing features to this POC-implementation as the (total) spent time ratio values are calculated dynamically instead of being stored in the DB as issue attribute values. As such am I interested in seeing how the Easy Redmine fellas have implemented this feature...

Some screenshots (note: please do not care about the other visible changes besides what's marked with green rectangles):

- issues list:

2021-08-13_19-25-33.png

- issue show:

2021-08-13_19-26-15.png

- version show:

2021-08-13_19-26-38.png

Any feedback is welcome...

**#3 - 2021-08-13 21:12 - Mischa The Evil**

*- File 0001-Add-a-basic-and-incomplete-POC-implementation-of-tot.patch added*

The previously attached patch has an issue. I'll hereby attach an updated patch.

**#4 - 2021-08-13 21:12 - Mischa The Evil**

*- File deleted (0001-Add-a-basic-and-incomplete-POC-implementation-of-tot.patch)*

**#5 - 2021-08-14 02:26 - Go MAEDA**

Mischa The Evil wrote:

> I'll attach a patch for demonstration and testing purposes only. This patch adds a basic and incomplete POC-implementation of (total) spent_time_ratio columns. It does not yet provide:
>
> - Column sorting functionality;

I think we don't have to implement the feature. Easy Redmine does not support column sorting. And it is useful enough without sorting.

**#6 - 2021-08-20 17:51 - Mischa The Evil**

Go MAEDA wrote:

> Mischa The Evil wrote:
>
> > [...]
> >
> > - Column sorting functionality;
>
> I think we don't have to implement the feature. Easy Redmine does not support column sorting. And it is useful enough without sorting.

I do not agree. When we introduce a column like this, I think we (eventually) want to sort by the value. And given the screenshot posted in this issues description, that implementation also supports column sorting (or it is using the same styling for sortable and not-sortable columns, which I can't really believe [and which I'd find bad UI/UX]).

Despite the above, I currently have the changes in a state where they do include:

- column sorting functionality, and
- filters for the added columns.

Testing this all manually, I came to a question: how to handle 0(%) values? And, more importantly, when to return them?
Up until now I decided to implement this in a way that 0.00% values where returned when an issue has an estimated time (independent of whether it has any spent time or not), and that the returned value is blank (nil) when an issue has no estimated time (again, independent of whether it has any spent time or not). This is visible in the screenshots I posted (which, as I stated previously, also included a change to not show 0:00 spent time values).
Now, with the above outlined implementation, the filters (and the column sorting, but that is not really a problem with an additional second sort clause) handle both values (0 and nil) equally. This is expected but might be confusing and a little bit ugly.

Before I spend more time on this I'd like some more feedback on how this should behave.

FTR: I'll post an updated patch for review including these features later on. I especially need a thorough review regarding the SQL that's included as that's not where I shine particularly...

**#7 - 2021-08-27 08:15 - Takenori TAKAKI**

*- File 0002-Filterable-and-Sortable-POC-implementation.patch added*

I think filtering and sorting spent_time_ratio will be a useful feature to know the status of the project.

I will post a proposed implementation of filterable and sortable spent_time_ratio using SQL.
tested it manually with postgres, mysql, and sqlite and confirmed that it works as it should be.

As a side note, If the filter for spent_time_ratio is set to "any", issues with a spent_time_ratio of 0% will be included in the results, just like the filter for estimated_time.

**#8 - 2021-09-05 22:56 - Marius BĂLTEANU**

*- Assignee set to Marius BĂLTEANU*

*- Target version set to Candidate for next major release*

## Files

| | | | |
|---|---|---|---|
| spent-time-ratio.png | 176 KB | 2021-08-01 | Go MAEDA |
| 2021-08-13_19-25-33.png | 29 KB | 2021-08-13 | Mischa The Evil |
| 2021-08-13_19-26-15.png | 18.7 KB | 2021-08-13 | Mischa The Evil |
| 2021-08-13_19-26-38.png | 3.53 KB | 2021-08-13 | Mischa The Evil |
| 0001-Add-a-basic-and-incomplete-POC-implementation-of-tot.patch | 11.1 KB | 2021-08-13 | Mischa The Evil |
| 0002-Filterable-and-Sortable-POC-implementation.patch | 7.77 KB | 2021-08-27 | Takenori TAKAKI |