redmineorg-copy202205 - Vote #66259

Add better presentation of issue status history

2022/05/09 14:23 - Admin Redmine

ステータス:	New	**開始日:**	2009/12/25
優先度:	通常	**期日:**	
担当者:		**進捗率:**	0%
カテゴリ:	UI_10	**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/4487	**status_id:**	1
category_id:	10	**tracker_id:**	2
version_id:	0	**plus1:**	0
issue_org_id:	4487	**affected_version:**	
author_id:	10437	**closed_on:**	
assigned_to_id:	0	**affected_version_id:**	
comments:	6		

説明

Currently, the only way to view the history of an issue is to scroll down the list of comments.
From a usability perspective, this is quite a tedious way to do it (see attachment: overview is a bit lacking).

It would be nice to have single overview of the issue history right below the ticket, for example near the "Associated revisions" section.

journals

What might such a summary look like?

I think there is good, deeper point on how issue history should be displayed on the issue.

It is true that if we see typically well discussed issue in Redmine, it is a huge trail of messages and across the life style. However, the presentation of how issue evolved (specially during specific phase) should be easily graspable.

Here are some concrete things we can do in the presentations:

h2. 1. Group notes outline based on status changes.

This is because, status is a fundamental outline the presentation could look like :

+*Status: Client Approval Phase*+ updated '100 days ago' by 'Alice'

Update by ...
" Note comes here"

Update by ...
" Note comes here"

+*Status: Development Phase*+ updated '70 days ago' by 'Bob'

Update by ...
" Note comes here"

Update by ...
" Note comes here"

+*Status: QA Phase*+ updated '20 days ago' by 'Danny'

```
Update  by  ...
"  Note  comes  here"

Update  by  ...
"  Note  comes  here"
```

---

h2. 2. Nested notes for replies

```
Update  by  ...
"  Note  comes  here"
    <pre>  Update  by  ...
      "  Note  comes  here"  </pre>
```

---

h2. 3. Distinguish Notes from updates

```
Issue  Update  by  ...
    Field:  x  ->  p
    Field:  y  ->  q

Notes  added  by  ...
"  Note  comes  here"
```

I think there can be many ideas on this. However, it is true that there is a strong need for improving outline of the issue page.

Add related #1834

Some more finer points:

1. Collapsible Phase Divs
One of the good reason why entire history should be divided in terms of phases (issue status transitions) -it's because issue status critically distinguish type of activities and related discussions. Also, we can have collapsible divs where one can hide the entire history and click to open only the phase you are interested in.
This way I can quickly jump to specific notes about 'What was discussed during specification phase' or 'what were QA comments during testing' etc.

2. Summarize issuse (again) on status transition
When many custom fields are used to mark all very specific issue when moves from one phase to another - it is worth while to capture the snapshot at that time. This way I don't have to hunt which fields changed when in what order.
I can then quickly identify what was the explanation of 'root cause' and 'resolution' before the issue moved to (or entered in) the testing phase.

3. List of associated people
Contributors, assigned, watchers with count. This can be on the top of the line of phase.

4. Tabs and/or filters
Different category of information can be divided in tab form or can be filtered to quickly find information we want. This division can be :

Notes,

Changes in custom fields

Time Tracking information

Associated revisions

Planning related fields (start-end dates and assignments)

5. High light things like priority changes

6. Simple graphical representations
Simple graphical representations might tell a good story - e.g.

how start-end dates have changed over time

number of comments or other issue activities over period of times

number of people worked over time and over different phases.

hours spent etc.

Different phases can be marked as regions on the timeline

Great ideas!

For muchbetter usability +1

**Dipan Mehta wrote:

> Some more finer points:
>
> *1. Collapsible Phase Divs*
> One of the good reason why entire history should be divided in terms of phases (issue status transitions) -it's because issue status critically distinguish type of activities and related discussions. Also, we can have collapsible divs where one can hide the entire history and click to open only the phase you are interested in.
> This way I can quickly jump to specific notes about 'What was discussed during specification phase' or 'what were QA comments during testing' etc.
>
> *2. Summarize issuse (again) on status transition*
> When many custom fields are used to mark all very specific issue when moves from one phase to another - it is worth while to capture the snapshot at that time. This way I don't have to hunt which fields changed when in what order.
> I can then quickly identify what was the explanation of 'root cause' and 'resolution' before the issue moved to (or entered in) the testing phase.
>
> *3. List of associated people*
> Contributors, assigned, watchers with count. This can be on the top of the line of phase.
>
> *4. Tabs and/or filters*
> Different category of information can be divided in tab form or can be filtered to quickly find information we want. This division can be :
>
> Notes,
>
> Changes in custom fields
>
> Time Tracking information
>
> Associated revisions
>
> Planning related fields (start-end dates and assignments)
>
> *5. High light things like priority changes*
>
> *6. Simple graphical representations*
> Simple graphical representations might tell a good story - e.g.
>
> how start-end dates have changed over time

number of comments or other issue activities over period of times

number of people worked over time and over different phases.

hours spent etc.

Different phases can be marked as regions on the timeline

---

related_issues

relates,New,12789,Redmine - design study
relates,New,1834,Threaded notes
relates,Closed,3058,Show issue history using tabs

履歴

#1 - 2022/05/10 17:24 - Admin Redmine

- カテゴリ を UI_10 にセット