

Redmine - Feature #6698

add insane set of features to subprojects

2010-10-19 20:20 - Albert Rosenfield

Status:	New	Start date:	2010-10-19
Priority:	Normal	Due date:	
Assignee:		% Done:	0%
Category:	Administration	Estimated time:	0.00 hour
Target version:			
Resolution:			

Description

Subprojects are phenomenal.

But I'd still like to see some features added :-).

- Ability to assign a custom subproject label to individual projects.
- Ability to list subprojects in custom order instead of alphabetically.
- Ability to duplicate an entire subproject structure by using a project as a template.
- Labels to complement subprojects.

(Sorry about the multiple items, it seemed necessary for creating a complete picture.)

Details

Custom labels (bullet №1)

...because: the "sub-something" nomenclature is confusing. A fix could be to let each project have two new properties called "subprojects are (singular):" and "subprojects are (plural):". The superuser could set these values to denote a proper name for subprojects within the context of the given project.

The labels would be used whenever the UI presents users with a concrete list of available subprojects to choose from.

(Default values: I(:field_subproject) and I(:label_subproject_plural).)

Custom ordering of subprojects (bullet №2)

...because this seems to be the only major (?) feature that timelog activities have which subprojects do not. If fixed, subprojects would be a suitable replacement for the timelog activity field, and all the confusing infrastructure surrounding that could be removed.

(Default sort order: depending on per-project "alphabetically sort subprojects" checkbox; either alphabetical or creation order.)

Use projects as templates (bullet №3)

...because projects have lots of options, and in order for projects/subprojects to replace other features, it must be super easy for superusers to create a new item based on something similar that is already there.

(Default value for project identifiers when duplicating a tree of projects/subprojects: "" (blank). The system could either use the project auto-increment ids as the project identifiers until the superuser comes around to assign a canonical identifier for each. Alternatively, the system could incorporate a filter to deny access to a new project until an identifier is assigned.)

Labels (bullet №4)

...because subprojects are only useful for stuff that you want to force people to decide a canonical location for. Such as which Customer an issue belongs to. Or whether a report is regarding a Bug/Feature/Patch ;-). Or which Activity a timelog entry should belong under.

A complementary feature, labels, would be useful for all other situations. When an administrator or superuser decides that it is perfectly acceptable for an issue to select multiple bindings for some particular matter, s/he would use labels rather than subprojects for that.

Google Code has an exemplary two-level label system, where a dash signifies the division into two levels in an otherwise ordinary text string. (Users can even create their own items on both levels by simply conjuring up a string. I'm not sure if this is a good or bad thing.)