Redmine - Feature #29758

Introduce optional pay per feature model of Redmine development

2018-10-11 23:58 - crypto gopher

Status:	New	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

What I'm proposing is a broad idea of moving to a model of development that can be actively supported by users.

Redmine is a very successful project of invaluable significance for people and organizations around the world. I'm not afraid to ascribe it a significant contribution to rising global effectiveness and letting people get their work done in ordered and prioritised manner.

Yet due to the nature of FOSS development it lacks power to provide some needed and long awaited features. Browsing through issue history makes you find multiple issues as old as 10 years. People again and again ask for crucial features, but except for adding "+1" there is no easy way they can support development.

Why not let users - of which some are companies - fund development time according to their needs?

I wrote proof of concept plugin that allows people to vote on Redmine issues by sending amount of cryptocurrencies. You can find it here: https://www.redmine.org/plugins/token-voting Be careful though: it is not something ready for production use. It requires further work.

I think that you should consider possibility of using such pay per feature model as an option for future development. I cannot predict whether it will work for Redmine (or even work for any project at all). But I think it is worth trying. In case you find the idea interesting I'm ready to work on this plugin further. As an long time Redmine user I feel the desire to give something back to the community.

Related issues:

Related to Redmine - Feature #13853: Is it possible to sponsor the implementa...	**Closed**

History

#1 - 2018-10-12 02:47 - Go MAEDA

- Related to Feature #13853: Is it possible to sponsor the implementation of features? added

#2 - 2018-10-13 01:54 - crypto gopher

Regarding the related #13853 I would like to emphasize crucial differences between idea explained here and the answer given in #13853#note-5

Solution proposed in #13853#note-5 mentions hiring Ruby consultant by interested entity (user or company) and going through the process of implementing feature by himself. While it is a valid solution it suffers at least these 2 **weaknesses**:

- the **cost of creating feature is placed on single entity**. The problem I see here is that many times cost (in terms of money, but also time required to go through the process) for the single entity can significantly exceed potential profits. Thus such solution may be economically unjustified.
- particular **needs of single entities that could finance development of some features may not overlap with the needs of wide community**. That's why I think that there is significant risk, that such code will not be accepted into Redmine. That in turn will encourage entities to either keep their code for themselves or put it into plugins. Problem with vital features implemented in plugin is that one cannot safely rely upon them in longer term (maintainability).

Proposed solution has following **advantages**:

- **allows to distribute cost between many interested entities**. Some of the most wanted features have tens of proponents. Just 10 of enities willing to support particular feature make entry barrier one order of magnitute smaller.
- besides providing financial incentive, **voting with tokens will provide information regarding popularity** of particular feature. This way developers can both optimize for most frequently wanted features and avoid implementing those least requested.
- **freedom of choice of what to implement and what not is retained** in the hands of Redmine developers. All votes on unimplemented features go back to voter sooner or later.
- most of the development **work can be done by competent people**, already knowing Redmine internals best.
- last but not least I predict it may introduce some **gamification** elements into development process. It may be not only rewarding, but also fun.