Redmine - Feature #14961

Reconsider moving from svn to git & GitHub

2013-09-21 16:11 - Łukasz Jąder

Status:	Reopened	**Start date:**	
Priority:	Low	**Due date:**	
Assignee:		**% Done:**	0%
Category:	Website (redmine.org)	**Estimated time:**	0.00 hour
Target version:			
Resolution:			

Description

Project could greatly benefit from ease of contributions from Pull Requests.

Issues would be on redmine.org but resolution would come in form of PR. Also GH has intergration hooks with redmine.

What are the downsides that stops this move to happen? Please provide some extended description of your decission.

Related issues:

Related to Redmine - Defect #11918: Official git / github mirror of redmine r...	**Closed**

History

#1 - 2013-09-21 18:13 - Toshi MARUYAMA

- Status changed from New to Closed

- Resolution set to Duplicate

#11918 has discussions.

As I described https://www.chiliproject.org/issues/601#note-32 ,
pull request is not suitable for big well-maintained centralized project.

#2 - 2013-09-21 18:14 - Toshi MARUYAMA

- Is duplicate of Defect #11918: Official git / github mirror of redmine repository is broken added

#3 - 2013-09-21 18:15 - Toshi MARUYAMA

- Status changed from Closed to Reopened

- Priority changed from High to Low

- Resolution deleted (Duplicate)

#11918 closed, so I reopen this issue.

#4 - 2013-09-21 18:16 - Toshi MARUYAMA

- Is duplicate of deleted (Defect #11918: Official git / github mirror of redmine repository is broken)

#5 - 2013-09-21 18:16 - Toshi MARUYAMA

- Related to Defect #11918: Official git / github mirror of redmine repository is broken added

#6 - 2013-09-21 18:16 - Toshi MARUYAMA

- Assignee deleted (Jean-Philippe Lang)

#7 - 2013-09-21 18:24 - Toshi MARUYAMA

Git is less functionality than Subversion and Mercurial.

- Git branch is **not stable**.
  http://mercurial.selenic.com/wiki/GitConcepts#Branch_model
- Git does not track file renaming.

#8 - 2013-09-21 19:25 - Łukasz Jąder

What do you mean by "Git branch is not stable"? Please provide some details.

I think Redmine is wonderfull software. It could be even more awesome if more people were involved in improving it.
At current state GitHub is de facto best way to share an opensource project and involve many people, beacause it is so easy to fix one thing and send Pull Request with the fix.

There are huge number of opensource projects hosted on GitHub where code is contributed in form of PRs. An it works very well for them. Jenkins CI is great example - as soon as project went GH and there was some maintainers group - the amount of features/plugins exploded and currently it is very powerfull tool.
Also many centralized companies buy 'GitHub Enterprise' just because of this feature - the ability to easy share your fix with people.

So I've began this discussion mostly because I'd like to see something simillar with Redmine - there are many places it already should and could be better (ex. Rest API) and there are people wanting to help.

And I thing switch to GH would greatly ease way for more people to bring there fixes and again bring 'the hype' to the project ;).

What do you think?

#9 - 2013-09-22 05:37 - Toshi MARUYAMA

Łukasz Jąder wrote:

> What do you mean by "Git branch is not stable"? Please provide some details.

```
$ git clone --bare git://github.com/chiliproject/chiliproject.git
Initialized empty Git repository in /REDMINE-1/git-workdir/chiliproject.git/
remote: Counting objects: 62920, done.
remote: Compressing objects: 100% (17081/17081), done.
remote: Total 62920 (delta 48160), reused 58446 (delta 44570)
Receiving objects: 100% (62920/62920), 18.59 MiB | 321 KiB/s, done.
Resolving deltas: 100% (48160/48160), done.

$ cd chiliproject.git/
$ git branch
* master
  release-v2.2.0
  release-v3.0.0
  release-v3.2.2
  stable
  stable-1.x
  stable-2.x
  unstable

$ git log -n 1 stable
commit 93de0ba668cb61621286930a522fd69efed9b24d
Author: Holger Just <XXXXXXX>
Date:   Tue Mar 19 22:36:48 2013 +0100

    Bump version to 3.8.0

$ git branch --contains 93de0ba668cb61621286930a522fd69efed9b24d
* master
  stable
  unstable

$ git branch -d stable
Deleted branch stable (was 93de0ba).

$ git branch --contains 93de0ba668cb61621286930a522fd69efed9b24d
* master
  unstable
```

#10 - 2013-09-22 05:46 - Toshi MARUYAMA

Łukasz Jąder wrote:

> At current state GitHub is de facto best way to share an opensource project and involve many people, beacause it is so easy to fix one thing and send Pull Request with the fix.
>
> There are huge number of opensource projects hosted on GitHub where code is contributed in form of PRs

GitHub is **not** de facto best way.

As I described https://www.chiliproject.org/issues/601#note-32 ,
Linus refused Pull Request.
https://github.com/torvalds/linux/pull/17

#11 - 2013-09-22 05:53 - Toshi MARUYAMA

Łukasz Jąder wrote:

> Also many centralized companies buy 'GitHub Enterprise' just because of this feature - the ability to easy share your fix with people.

GitHub is one of Git hosting service.
Bitbukcet, CodePlex, Google Code, Souceforge provides hosting service.
GitHub provides **only Git**.

Many centralized companies use own repositories in intranet, not use hosting services.

#12 - 2013-09-22 18:38 - Łukasz Jąder

What do you mean by sample in comment 9 - that you can delete branch with name 'stable'? I don't get it.

I've read discussion in PR https://github.com/torvalds/linux/pull/17 and I completly agree with Linus arguments. Linux and Git projects have certain coding and commit standards, which Linus wants to keep. He also thinks that those standards are objectively better so other people would benefit from adhering to them. But he also accepts that other people decide to comply with other set of rules for their projects.

That being said I think that Redmine project when moving to GitHub should also provide its set of rules of contribution. They could be exactly the same as Linus ones, I'm fine with that.

Refering to:

> Many centralized companies use own repositories in intranet, not use hosting services.

You could also use GitHub Enterprise in intranet - there is such option, but its costs more.

#13 - 2013-09-22 18:39 - Łukasz Jąder

Refering to your comment https://www.chiliproject.org/issues/601#note-32 :

> pull request is not suitable for big well-maintained centralized project.

I think PRs works quite good with projects in all sizes and technologies. Great acknowledgement of that are many open source projects on GitHub:

- bootstrap
- jekyll
- gitlabhq
- octopress
- diaspora
- angular.js

They are quite big projects and with developing them with GH works quite well.

What I ask is a little discussion about feature of Redmine project and evolution of its rules regarding opensource contributions.

So here are some pros and cons (with my comment) of this move:
Pros:
- move to GH would involve much more people who are willing to help
- projects on GH gets more attention and generally more usage which provide more feedback/testing
- more people invlolved means more interesting ideas/use cases
- people love attribution for their work - and with GH it is clear and open that someone worked with some part of project - which is awesome ego boost
- as in Jenkins - GitHub PRs could be tied with issues in Issue tracker, and discussion could be managed only on redmine.org
- git in general is easier to work with and more performant than svn
- with good contribution standards it is clear what conditions should one's PR shold have
- only certain group of maintainers have commit rights to direct commit main repo; other people work is aproved or rejected with some meaningful comment.

Cons:
- move to GH forces on every people to change their habbits (changes many times are good things ;))
- more attention means more 'noise' and more effort with managing/reviewing contributions
- there will be many poor quality contributions and it would take more time to manage

I'm open for discussion :).

#14 - 2013-09-23 01:10 - Toshi MARUYAMA

Łukasz Jąder wrote:

> What do you mean by sample in comment 9 - that you can delete branch with name 'stable'? I don't get it.
>
> I've read discussion in PR https://github.com/torvalds/linux/pull/17 and I completly agree with Linus arguments. Linux and Git projects have certain coding and commit standards, which Linus wants to keep. He also thinks that those standards are objectively better so other people would benefit from adhering to them. But he also accepts that other people decide to comply with other set of rules for their projects.

Have you read our Contributing guide?

> That being said I think that Redmine project when moving to GitHub should also provide its set of rules of contribution. They could be exactly the same as Linus ones, I'm fine with that.
>
> Refering to:
>
> > Many centralized companies use own repositories in intranet, not use hosting services.
>
> You could also use GitHub Enterprise in intranet - there is such option, but its costs more.

I cannot understand benefit of moving Github than our Contributing guide.

#15 - 2013-09-23 01:19 - Toshi MARUYAMA

Łukasz Jąder wrote:

> Refering to your comment https://www.chiliproject.org/issues/601#note-32 :
>
> > pull request is not suitable for big well-maintained centralized project.
>
> I think PRs works quite good with projects in all sizes and technologies. Great acknowledgement of that are many open source projects on GitHub:
> ...

I disagree all of your this comment.
We use CI server and Covarage on **redmine.org**.

Does GitHub provide CI and Coverage services **on own host**?

If GitHub stopped service suddenly, such as Google Reader, we lost all resources.
I cannot accept this risk.

#16 - 2013-09-23 01:30 - Toshi MARUYAMA

Łukasz Jąder wrote:

> What do you mean by sample in comment 9 - that you can delete branch with name 'stable'? I don't get it.

Subversion and Mercurial branch tie to **one** branch.
r12156 is trunk, r12160 is 2.3-stable.

Git branch is the reffernce to revisions.
You need to run "git branch --contains revision" in order to know "What branch is this revision in".

If you delete Git branch, you lose informaition of "What branch is this revision from"?

#17 - 2013-09-23 11:31 - Łukasz Jąder

> Have you read our Contributing guide?
> I cannot understand benefit of moving Github than our Contributing guide.

Yes I've read it and I raise this issue as a request to revise it :).
I have strong concerns that Redmine project is losing people attention and willingness to help and I thing switch to git/GH would prevent this. I've scanned through activity page since last month an there are aprox. 40 changes involved with issues with .patches. From this 40, 20 are patches at least one year old and probably require little work with them. For such big app like Redmine with GitHub I hope amount of new patches would at least double :).

My point is: with GH project would much more others dev attention and would recive much more code changes in form of PRs. I think it is a good thing to be able to have new features more quicky and from my observations GH hosted projects evolve quicker than others.

Jenkins CI has alse quite nice standard that plugins for Jenkins are hosted in organisation 'jenkinsci'. Original plugin authors recive commit access to this plugin repo and began work as its maintainers. With this approach plugins are easy discoverable and people know where to send PRs to. When plugin maintainer wants to leave its role, other people can step in.

> Does GitHub provide CI and Coverage services on own host?

All I ask is to move code to Git and GitHub and accept work from others as PRs. I don't want to move issue managment there.
Issues should be reported here, but instead of submittig patches people would provide PRs.

> If GitHub stopped service suddenly, such as Google Reader, we lost all resources.
> I cannot accept this risk.

The good thing with git is that you can't loose history because every fork has complete history tree. So if GH stopped its service you could simply push entire repository to different location :). Or use your own hosted git repo with ability to manage PRs, ex. gitlab or rhodecode (i they had free OS version).

> Git branch is the reffernce to revisions.
> You need to run "git branch --contains revision" in order to know "What branch is this revision in".
> If you delete Git branch, you lose informaition of "What branch is this revision from"?

In git you can assume/require convention that 'master' == 'trunk'. Why do you need to ask question "What branch is this revision in"? To backport some fixes to stable releases? If so then maybe you should consider working with git where with every PR there is 'merge' commit (--no-ff merge) - many projects work this way. Then in master you have all markers of merged branches and backporting to stable releases would require exact same merge --no-ff of PR related branch.

I'm also curious how other Redmine maintainers are seeing this proposal? Do you have contact with them, so you can ask?

Thanks for your time to have this discussion.

#18 - 2013-09-23 12:56 - Toshi MARUYAMA

Łukasz Jąder wrote:

> > Have you read our Contributing guide?
> > I cannot understand benefit of moving Github than our Contributing guide.
>
> Yes I've read it and I raise this issue as a request to revise it :).
> I have strong concerns that Redmine project is losing people attention and willingness to help and I thing switch to git/GH would prevent this. I've scanned through activity page since last month an there are aprox. 40 changes involved with issues with .patches. From this 40, 20 are patches at least one year old and probably require little work with them. For such big app like Redmine with GitHub I hope amount of new patches would at least double :).

Our commit rule is "Bug fix and adding feature should have **tests**".
We accept patches with this rule.
It does not related Pull request or Patches.

> My point is: with GH project would much more others dev attention and would recive much more code changes in form of PRs. I think it is a good thing to be able to have new features more quicky and from my observations GH hosted projects evolve quicker than others.
>
> Jenkins CI has alse quite nice standard that plugins for Jenkins are hosted in organisation 'jenkinsci'. Original plugin authors recive commit access to this plugin repo and began work as its maintainers. With this approach plugins are easy discoverable and people know where to send PRs to. When plugin maintainer wants to leave its role, other people can step in.

We don't care about plugins.
Some plugins have own tests.

- https://github.com/backlogs/redmine_backlogs
  - https://travis-ci.org/backlogs/redmine_backlogs

> > Does GitHub provide CI and Coverage services on own host?
>
> All I ask is to move code to Git and GitHub and accept work from others as PRs. I don't want to move issue managment there.
> Issues should be reported here, but instead of submittig patches people would provide PRs.
>
> > If GitHub stopped service suddenly, such as Google Reader, we lost all resources.

>> I cannot accept this risk.
>
> The good thing with git is that you can't loose history because every fork has complete history tree. So if GH stopped its service you could simply push entire repository to different location :). Or use your own hosted git repo with ability to manage PRs, ex. gitlab or rhodecode (i they had free OS version).

I know all revisions can be mirrored.
I mean that **all discussion** should be on **redmine.org**.
So, for code review, we should use codereview plugin instead of other hosting service.
http://www.redmine.org/plugins/codereview

#19 - 2013-09-23 13:17 - Toshi MARUYAMA

- File chili-ecb29a600b5.png added

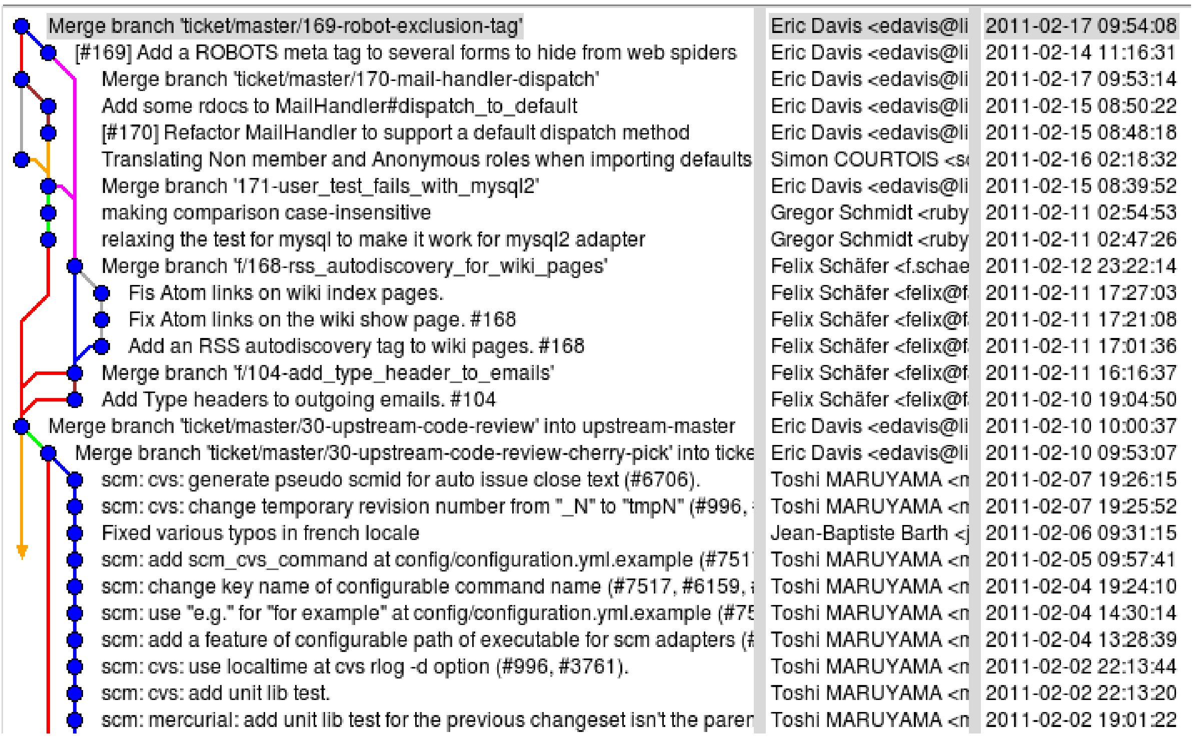
- File chili-de-gh-network.png added

Łukasz Jąder wrote:

>> Git branch is the reffernce to revisions.
>> You need to run "git branch --contains revision" in order to know "What branch is this revision in".
>> If you delete Git branch, you lose informaition of "What branch is this revision from"?
>
> In git you can assume/require convention that 'master' == 'trunk'. Why do you need to ask question "What branch is this revision in"? To backport some fixes to stable releases? If so then maybe you should consider working with git where with every PR there is 'merge' commit (--no-ff merge) - many projects work this way. Then in master you have all markers of merged branches and backporting to stable releases would require exact same merge --no-ff of PR related branch.

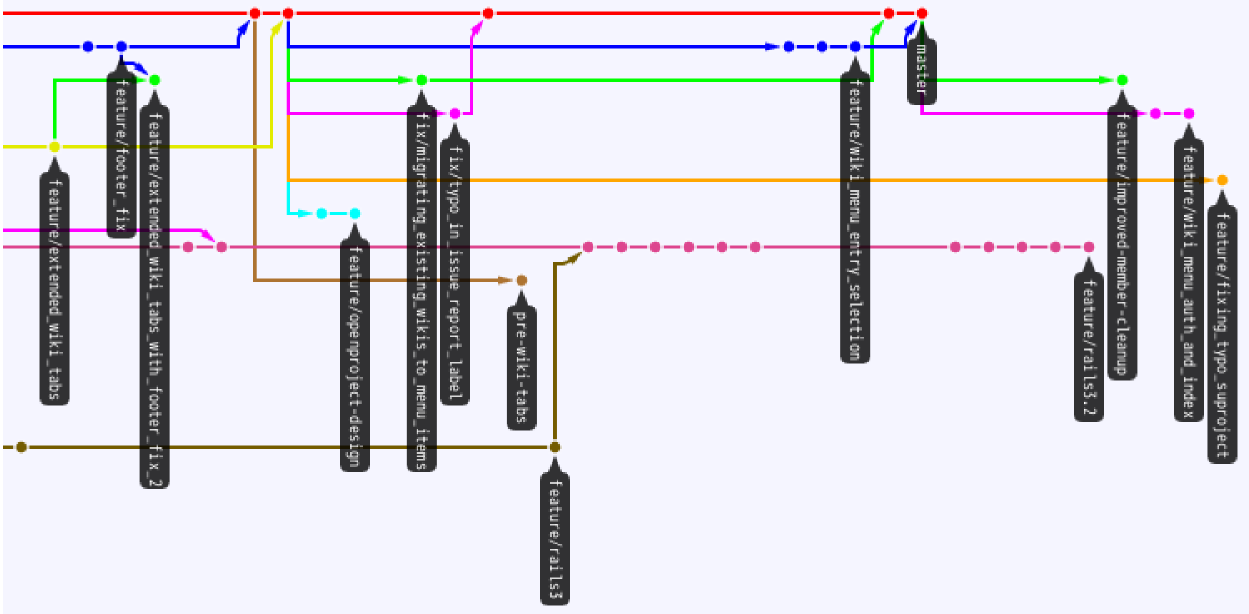
I dislike "--no-ff merge".
Because revision graph becomes very dirty.

As I described my Japanese presentation and https://www.chiliproject.org/issues/127#note-8 ,
main line should be **rebased**.



Merge branch 'ticket/master/169-robot-exclusion-tag'	Eric Davis <edavis@li	2011-02-17 09:54:08
[#169] Add a ROBOTS meta tag to several forms to hide from web spiders	Eric Davis <edavis@li	2011-02-14 11:16:31
Merge branch 'ticket/master/170-mail-handler-dispatch'	Eric Davis <edavis@li	2011-02-17 09:53:14
Add some rdocs to MailHandler#dispatch_to_default	Eric Davis <edavis@li	2011-02-15 08:50:22
[#170] Refactor MailHandler to support a default dispatch method	Eric Davis <edavis@li	2011-02-15 08:48:18
Translating Non member and Anonymous roles when importing defaults	Simon COURTOIS <s	2011-02-16 02:18:32
Merge branch '171-user_test_fails_with_mysql2'	Eric Davis <edavis@li	2011-02-15 08:39:52
making comparison case-insensitive	Gregor Schmidt <ruby	2011-02-11 02:54:53
relaxing the test for mysql to make it work for mysql2 adapter	Gregor Schmidt <ruby	2011-02-11 02:47:26
Merge branch 'f/168-rss_autodiscovery_for_wiki_pages'	Felix Schäfer <f.schae	2011-02-12 23:22:14
Fis Atom links on wiki index pages.	Felix Schäfer <felix@f	2011-02-11 17:27:03
Fix Atom links on the wiki show page. #168	Felix Schäfer <felix@f	2011-02-11 17:21:08
Add an RSS autodiscovery tag to wiki pages. #168	Felix Schäfer <felix@f	2011-02-11 17:01:36
Merge branch 'f/104-add_type_header_to_emails'	Felix Schäfer <felix@f	2011-02-11 16:16:37
Add Type headers to outgoing emails. #104	Felix Schäfer <felix@f	2011-02-10 19:04:50
Merge branch 'ticket/master/30-upstream-code-review' into upstream-master	Eric Davis <edavis@li	2011-02-10 10:00:37
Merge branch 'ticket/master/30-upstream-code-review-cherry-pick' into ticke	Eric Davis <edavis@li	2011-02-10 09:53:07
scm: cvs: generate pseudo scmid for auto issue close text (#6706).	Toshi MARUYAMA <n	2011-02-07 19:26:15
scm: cvs: change temporary revision number from "_N" to "tmpN" (#996,	Toshi MARUYAMA <n	2011-02-07 19:25:52
Fixed various typos in french locale	Jean-Baptiste Barth <j	2011-02-06 09:31:15
scm: add scm_cvs_command at config/configuration.yml.example (#751	Toshi MARUYAMA <n	2011-02-05 09:57:41
scm: change key name of configurable command name (#7517, #6159,	Toshi MARUYAMA <n	2011-02-04 19:24:10
scm: use "e.g." for "for example" at config/configuration.yml.example (#75	Toshi MARUYAMA <n	2011-02-04 14:30:14
scm: add a feature of configurable path of executable for scm adapters (#	Toshi MARUYAMA <n	2011-02-04 13:28:39
scm: cvs: use localtime at cvs rlog -d option (#996, #3761).	Toshi MARUYAMA <n	2011-02-02 22:13:44
scm: cvs: add unit lib test.	Toshi MARUYAMA <n	2011-02-02 22:13:20
scm: mercurial: add unit lib test for the previous changeset isn't the paren	Toshi MARUYAMA <n	2011-02-02 19:01:22

Revision graph of GitHub pull request workflow is very dirty.


feature/extended_wiki_tabs
feature/footer_fix
feature/extended_wiki_tabs_with_footer_fix_2
feature/openproject-design
fix/migrating_existing_wikis_to_menu_items
fix/typo_in_issue_report_label
pre-wiki-tabs
feature/rails3
feature/wiki_menu_entry_selection
master
feature/rails3.2
feature/improved-member-cleanup
feature/wiki_menu_auth_and_index
feature/fixing_typo_suproject

#20 - 2013-09-24 04:05 - rm user

> So, for code review, we should use codereview plugin instead of other hosting service.

http://www.redmine.org/plugins/codereview

I don't think this plugin gives enough functionality for code review.

It basically creates a blank Issue per each line or per commit and you can write comments in the Issue.

It's very basic and not really user-friendly, very complicated to use it for code review because you need to specify each time subject and multiple options per line and you can't review multiple lines at the same time.

GIT & SVN have their own drawbacks each other.

But it's proven that GIT's workflow is superior faster comparing to SVN, so having redmine on github.com would speed up things I'm pretty sure (mainly because github is an easy-to-use system).

I noticed main drawback in redmine is its plugins are not consistent so if there is an update coming of major version most likely all plugins will stop working and their authors not updating them before someone asks them to do so.

So if there will be redmine's official repository on github.com with all latest changes a lot of developers would benefit from this just by watching redmine's repository for changes.

Also its odd but git's support in redmine is still not perfect without 3rd party plugin (we're using redmine_git_hosting).

Standard support is very basic and not gives an ability to use/define SSH-keys for access as well as other things implemented in the 3rd party plugin.

Maybe it's because main developers are very conservative and like to stick to SVN.

#21 - 2013-09-24 05:41 - Toshi MARUYAMA

rm user wrote:

> Maybe it's because main developers are very conservative and like to stick to SVN.

No, we are not conservative.

Our rule is that **all bug fixes and features should have tests**.
It equals that **coverage should be near 100%**.
Due to this rule, we have ported Rails3 on 2012-05-15 (r9698).

ChiliProject and OpenProject have not ported Rails3 yet.

- https://www.chiliproject.org/issues/601
- OpenProject news : The next step: OpenProject on Rails 3.2 and Rails Engines

> But it's proven that GIT's workflow is superior faster comparing to SVN, so having redmine on github.com would speed up things I'm pretty sure (mainly because github is an easy-to-use system).

If these are true that "GIT's workflow is superior faster comparing to SVN" and "github is an easy-to-use system",
why have not ChiliProject and OpenProject ported Rails3 yet?

> So if there will be redmine's official repository on github.com with all latest changes a lot of developers would benefit from this just by watching redmine's repository for changes.

You can watch subversion commits by RSS.
http://www.redmine.org/projects/redmine/repository/revisions.atom

And there are mirrors on GitHub and Bitbucket.

> I noticed main drawback in redmine is its plugins are not consistent so if there is an update coming of major version most likely all plugins will stop working and their authors not updating them before someone asks them to do so.

It is not related with Redmine uses git or not.
Well-maintained plugins have tests.
Code review plugin uses Mercurial.
it has tests.
https://bitbucket.org/haru_iida/redmine_code_review/src/c9ad7f953c61f6b0dd60ab14f7d74551c8323b94/test?at=default
And tests run on Jenkins.
http://www.r-labs.org/projects/r-labs/hudson/index

#22 - 2013-09-24 06:08 - rm user

Toshi MARUYAMA wrote:

> rm user wrote:
>
>> Maybe it's because main developers are very conservative and like to stick to SVN.
>
> No, we are not conservative.
>
> Our rule is that **all bug fixes and features should have tests**.
> It equals that **coverage should be near 100%**.
> Due to this rule, we have ported Rails3 on 2012-05-15 (r9698).

That's very good. Do not get me wrong I'm glad that you guys are putting so much effort into this redmine project.

I'm just saying my own personal thoughts about git, because I've used svn for quiet some time.
Git is uber fast at file operation and clone.

> ChiliProject and OpenProject have not ported Rails3 yet.
>
> - https://www.chiliproject.org/issues/601
> - OpenProject news : The next step: OpenProject on Rails 3.2 and Rails Engines
>
>> But it's proven that GIT's workflow is superior faster comparing to SVN, so having redmine on github.com would speed up things I'm pretty sure (mainly because github is an easy-to-use system).
>
> If these are true that "GIT's workflow is superior faster comparing to SVN" and "github is an easy-to-use system",
> why have not ChiliProject and OpenProject ported Rails3 yet?

I think it's incorrect comparison, ChilliProject is going to be soon a standalone tracking system, they're changing a lot of things (comparing to Redmine) from what I heard.

Also I didn't say github is so good that you guys must switch immediately there.

I've said GIT's workflow not github's.. What I dislike about github that it's closed source and you can't use it as a redmine (i.e. on your own server with customizations).

>> So if there will be redmine's official repository on github.com with all latest changes a lot of developers would benefit from this just by watching redmine's repository for changes.
>
> You can watch subversion commits by RSS.

> http://www.redmine.org/projects/redmine/repository/revisions.atom
>
> And there are mirrors on GitHub and Bitbucket.

That's true, but nobody seems to watch pull requests on bitbucket and github mirrors. That should be accounted too, because not everyone is familiar with redmine and how to post here.

> It is not related with Redmine uses git or not.
> Well-maintained plugins have tests.
> Code review plugin uses Mercurial.
> it has tests.
> https://bitbucket.org/haru_iida/redmine_code_review/src/c9ad7f953c61f6b0dd60ab14f7d74551c8323b94/test?at=default
> And tests run on Jenkins.
> http://www.r-labs.org/projects/r-labs/hudson/index

That's good. Yes, probably you're right, I was just saying my own impression about Redmine and it's plugins. Usually have problems If I update redmine I have to hack plugins myself to make them work or submit issues over developers. Would be good to have an ability to link on Redmine's Plugins WIKI list actual plugin to the github/bitbucket repository to monitor and update changes based on history there..

#23 - 2013-09-26 22:30 - Gabriel Mazetto

Thanks god you guys are discussing Git vs SVN... I have something that could help move the codebase to Github and make everybody happy:

https://help.github.com/articles/support-for-subversion-clients

Yes, you can still use SVN and host it on Github and let everybody clone it with Git and make pull requests.
By moving to Github I would like to see travis and all the other cool services integrated for running continuos integration, code coverage and so on...
This is something that could move Redmine from "stagnated project" where only 4 ou 6 developers create code to something that more and more users could submit patches and improve the codebase.

Don't get me wrong, the burocracy for creating a patch file and hope someone see that the ticket has a patch and then mark it as having a patch, and then waiting for the eternity to have someone look at it and suggest changes to have to generate another patch file and everything again makes everyone who would like to give a contribution think twice.

By going the github way, you earn Travis integration directly to the pull requests, and this is a huge improvement by itself.

A similar project that is using github correctly is Gitlab (you should check the project and how the deal with it)... there is ton of contribution and all of this is thanks to github. Another example that failed for not using Github's power is Gitorious... they have a burocratic process and because of that, stagnated.

#24 - 2013-09-27 05:26 - Toshi MARUYAMA

- File openproject-travis.png added

Gabriel Mazetto wrote:

> By going the github way, you earn Travis integration directly to the pull requests, and this is a huge improvement by itself.

Again, we have own CI server on redmine.org.
http://www.redmine.org/builds/index.html

Travis is very slow.

https://travis-ci.org/opf/openproject/builds
OpenProject test costs three hours per one test.

openproject-travis.png
OpenProject tests on **only** MySQL and PostgreSQL and Ruby 1.9.
https://github.com/opf/openproject/blob/a1e67dd4607ce9/.travis.yml#L30

Redmine tests on many databases and Rubies.

Travis tests runs **all pull requests**.
It means "main line" branch tests wait pull request tests finishing.
It is stupid.

- https://github.com/opf/openproject/pull/246
- https://github.com/chiliproject/chiliproject/pull/278#issuecomment-20997479

#25 - 2013-10-07 00:19 - Gabriel Mazetto

I can see your point, and I'm not suggesting to stop using your own solution if it fits better, but travis makes possible to everyone who clones the project benefits from automated testings... (which can be enabled without any drawbacks)

I know chiliproject is controversial here, while they failed to keep the project going on, I can see that that happened more because of other reasons then the choices they made.

I do respect both teams as you guys did an awesome job creating and sustaining the project for so long, BUT for many years this project failed to increase the number and the quality of the contributions, and wasted many newcomers by the rigid statements against any sort of flexibilization on the way the project is managed.

While you may have your reasons and they may seen valid and state of art for you, sometimes dealing with community is all about being flexible aiming sustainable growth.

I've seen this type of thread for many years where thousands came here everyday and politely ask to have a more github-friendly workflow, like they being able to send patches using the "github.com/redmine/redmine" project, and as you can see by visiting redmine and gitlab's repositories, the amount of contributors who spend some time trying to help both projects is unquestionably bigger on the gitlab side.

This may be the last time I try to point this out: by failing to hear and adapt from other developers suggestions you are keeping the "contributors" list shorter and shorter.

As a way to make merge requests from github possible and keep you using SVN, you can grab a patchfile directly from the merge request: https://help.github.com/articles/merging-a-pull-request

#26 - 2013-10-07 01:20 - Toshi MARUYAMA

Gabriel Mazetto wrote:

> I can see your point, and I'm not suggesting to stop using your own solution if it fits better, but travis makes possible to everyone who clones the project benefits from automated testings... (which can be enabled without any drawbacks)

But, I will not manage .travis.yml and some rake tasks.
There is no melit for us.

> I know chiliproject is controversial here, while they failed to keep the project going on, I can see that that happened more because of other reasons then the choices they made.

This is the statement about OpenProject forking.
https://www.chiliproject.org/boards/1/topics/2247

But, biggest reason of OpenProject forking is that Eric refused whole revisions of Finnlab pull request.
https://github.com/chiliproject/chiliproject/pull/122

> I do respect both teams as you guys did an awesome job creating and sustaining the project for so long, BUT for many years this project failed to increase the number and the quality of the contributions, and wasted many newcomers by the rigid statements against any sort of flexibilization on the way the project is managed.

Again, we accecpt patches **with tests**.
It is not related pull requests or patches.

There are many pull requests for ChiliProject.
But, there is no test.
So, ChiliProject cannot accept pull request.
https://github.com/chiliproject/chiliproject/pulls

#27 - 2016-10-25 19:22 - Warren Postma

I'm a huge Gitlab fan and I'd like to see RedMine move to Git and be hosted on Gitlab.

I'm personally using Redmine for all my bugtracking and interested in working on improving Gitlab+Redmine integration.

It would hugely simplify the Redmine project if you could move to a Git and Merge Request workflow instead of logging issues with .patch files, which feels barbaric.

#28 - 2016-10-25 19:25 - Warren Postma

Are project maintainers in 2016 **REALLY** actually happy with Subversion? I personally think Subversion is junk. I have used it a lot, it's not that I don't know it. I just ditched it for Mercurial ages ago, and recently decided that the whole world has moved to Git, so I did too.

And having per-branch-CI automatically in gitlab would be a huge move forward. Are the main Redmine devs really seriously happy in Subversion?

Instead of having a README.md that says "Don't send pull requests", why not actually start accepting pull requests, but if Github is a bit spare (and I think it is), I think Gitlab.com would be much much better.

#29 - 2016-10-25 20:12 - Toshi MARUYAMA

Warren Postma wrote:

> the whole world has moved to Git,

I don't think so.
http://www.theregister.co.uk/2016/10/18/facebook_mercurial_devs_forget_git/

> "Initially, a lot of developers were skeptical about Mercurial and preferred Git," Szorc said. "Now, apparently a number of their developers have forgot how to use Git."

#30 - 2019-01-30 00:53 - Yasukazu Nagatomi

+1

#31 - 2019-01-30 06:44 - Anonymous

1+ for the Git, the pile of +1s is not big enough here, let's take +1s on this issue to the whole another level, if you already +1ed here, don't hesitate to +1 again! :D

Toshi MARUYAMA wrote:

> pull request is not suitable for big well-maintained centralized project.

IMHO, this project doesn't really fall into the category of well-maintained projects anymore, let's not run from the truth. It so seems like the reason to it is because it's too centralized actually,
and the central force responsible for reviewing the code and making it into the release is almost dead. The project is making extremely pathetic steps forward because of this.

Puzzle: what did dinosaurs do?
The correct answer: they extinct by the well known reasons!

IMHO the solution would be de facto to try to either decentralize it, which is via Git and PRs (hey, bitcoin is not that bad either after all) or at least have new central force responsible for core review and releases who could be a lot more active.

But sitting still and not admitting the extreme passiveness of evolution of this project, is the worst that can be done right now. Taking action to try to improve this would be better than not taking any action.

Please also see the issue from Marius: #30069

#8363 and #30069 are also probably related judging by the discussions that occurred here, please add as related if you think they are.

#32 - 2019-01-30 15:35 - Anonymous

Hello dinosaurs,

5 years have passed since this issue was made. I do not know if the whole world has moved to Git, but OSS Projects has become mainstream to collect contribution by Pull Requests and Git.

We might implement Pull Request on Redmine, but this will not be easy.

First of all, we should move Redmine's repository to Git and allow Pull Requests to be accepted. Then, we will lower the contribution hurdles and open the contribution gate widely. Even with using competing products.

Fortunately GitLab has the "External issue tracker". With this, you can continue to use many issues of Redmine.org as it is, while using GitLab's Git management and Pull Request (Merge Request).

Would you like to start discussing this issue again?

#33 - 2019-01-30 17:51 - Go MAEDA

I hope you all to be gentlemanly even if you don't like the current situation. You cannot improve Redmine by discouraging people who have been spending time on developing Redmine for years.

#34 - 2019-01-30 17:53 - rm user

Max Johansson wrote:

> IMHO, this project doesn't really fall into the category of well-maintained projects anymore, let's not run from the truth. It so seems like the reason to it is because it's too centralized actually,

and the central force responsible for reviewing the code and making it into the release is almost dead. The project is making extremely pathetic steps forward because of this.

Well, many projects out there are centralized and even those at github.
And it's not particully true regarding 'dead' - as new releases are keep coming here: https://www.redmine.org/projects/redmine/wiki/Download

Redmine's team is just very conservative and don't want to change what's already working for them.

Taiki I wrote:

> Hello dinosaurs,
>
> 5 years have passed since this issue was made. I do not know if the whole world has moved to Git, but OSS Projects has become mainstream to collect contribution by Pull Requests and Git.

You can't do much regarding this, there are many issues in redmine sitting there for much longer time.

However, there is a mirror in github of whole redmine repository here: https://github.com/redmine/redmine

And there are actually some pull requests! However they are not accepted there at all as you need to submit your pull request via different way..

> We might implement Pull Request on Redmine, but this will not be easy.

I agree Pull Requests mechanism should be there it's much easier to review Pull Requests in gitlab vs redmine and comment on certain code lines or/and accept or reject them.

> First of all, we should move Redmine's repository to Git and allow Pull Requests to be accepted. Then, we will lower the contribution hurdles and open the contribution gate widely. Even with using competing products.
>
> Fortunately GitLab has the "External issue tracker". With this, you can continue to use many issues of Redmine.org as it is, while using GitLab's Git management and Pull Request (Merge Request).

Gitlab can be used together with redmine just fine and there will be cross-link between the gitlab and redmine if you specify in a commit e.g: fixes #xxxx it will be automatically linked to your redmine.org issue.

> Would you like to start discussing this issue again?

I don't think there will be any changes in this area. Redmine devs are happy with SVN it all works for them, why change?

#35 - 2019-01-30 21:04 - Anonymous

rm user wrote:

> And it's not particularly true regarding 'dead' - as new releases are keep coming here: https://www.redmine.org/projects/redmine/wiki/Download

Never said it's dead, I said it's "almost dead" and I'm perfectly aware that it's still releasing once in a long while.

> Redmine devs are happy with SVN it all works for them, why change?

We don't know for sure yet, the opinion might have changed for majority of core devs, considering that there is only 2 maybe 3 of them consistently active now. The rest are mostly contributors, which more than likely wouldn't mind git with ability to pr. Maybe it would be best to do some kind of poll for this first actually.

#36 - 2019-01-30 21:22 - rm user

Max Johansson wrote:

> rm user wrote:
>
> > And it's not particularly true regarding 'dead' - as new releases are keep coming here: https://www.redmine.org/projects/redmine/wiki/Download
>
> Never said it's dead, I said it's "almost dead" and I'm perfectly aware that it's still releasing once in a long while.

I agree that there should be certain release cycle, e.g. release a new version every 2-3 months.

> > Redmine devs are happy with SVN it all works for them, why change?
>
> We don't know for sure yet, the opinion might have changed for majority of core devs, considering that there is only 2 maybe 3 of them consistently active now. The rest are mostly contributors, which more than likely wouldn't mind git with ability to pr. Maybe it would be best to do some kind of poll for this first actually.

There are 2 or 3 people who are commiting to the central SVN repository as project maintainers, but that doesn't mean that others are not contributing.

Well, you actually can name them here:
Jean-Philippe Lang
Go MAEDA
Toshi MARUYAMA (lately not active as much)

It's not that bad actually that there is a review process and only certain people are allowed to commit new changes.

From my own experience of using Readmine it's very rock solid software especially if you're on the stable branch, most issues are either caused by 3rd party plugins or by major updates (e.g. jumping from version 3.x to version 4.x) but it's understandable.

What I've always liked that redmine provided solid core software which can be extended in many ways, althrough not everyone wants to do that work on extension and want to get everything 'out of the box'.

Regarding, git integration I do agree it should be there and as a start DEVs could look into making their own gitlab instance on some virtual/physical server as a start or if they don't want to pay that extra for another server they could sign-up at gitlab.org and make public project for everyone. Code review and patches will be much more simplified for others instead of creating new issue they could use PR functionality and write the description there.

I also think the project is undermanned that's why you can see there is a significant pile of old issues still sitting, would be nice if DEVs could recruit some new project maintainers to help with old issues and tag them at least so similar ones will be linked to each other.

#37 - 2022-11-14 12:01 - Vincent Robert

As a developer of many plugins, I know I would have done a lot more contributions to Redmine Core if it was easier.
It may be a good time to reconsider moving from SVN, simplify processes and help new members to join the Redmine community.

Files

chili-de-gh-network.png	30.8 KB	2013-09-23	Toshi MARUYAMA
chili-ecb29a600b5.png	122 KB	2013-09-23	Toshi MARUYAMA
openproject-travis.png	70.9 KB	2013-09-27	Toshi MARUYAMA