

## ChiliProject - Task # 118: Scope and release plan 2.0.0

<b>Status:</b>	Closed	<b>Priority:</b>	Normal
<b>Author:</b>	Eric Davis	<b>Category:</b>	ChiliProject - Organization
<b>Created:</b>	2011-02-01	<b>Assignee:</b>	Eric Davis
<b>Updated:</b>	2011-07-04	<b>Due date:</b>	
<b>Remote issue URL:</b>			
<b>Description:</b> We need to decide what major features should go into 2.0.0. This will be the first major release and can break backwards compatibility with 1.x.  Ideally we would have 3-5 major features that are fully tested.  See also [[Important Features]] and [[Ideas]]			

### History

---

#### 2011-02-01 07:15 am - Eric Davis

A few ideas I propose (and have seen proposed)

- \* Tim's journalized patch
- \* New layout and theme based on the work I've done for Shane and Peter
- \* Complete REST API (will probably need a few client libraries built too as well as an audit of the existing API)
- \* Tagging
- \* More custom field types - multi-select lists, urls, associated data (list of users, list of projects, etc)

What other ideas are on the top of everyone's list?

#### 2011-02-01 11:49 am - Tim Felgentreff

MobileCSS. See #116.

#### 2011-02-01 11:51 am - Tim Felgentreff

Plugin-dependencies. See <http://www.redmine.org/issues/6324>

#### 2011-02-01 06:00 pm - Eric Davis

- *Target version deleted (1.1.0 "Bell")*

After thinking about this, I think it would be better to have people actually commit to doing something. We all have features we want and there are hundreds of ideas but unless someone actually does the work for the feature, it won't get done. So I propose commenting like:

- \* Eric - I will add support for XYZ Widgets (#0)
- \* Eric - I will audit and review the ABC class to make it compatible with DEF.com

#### 2011-02-01 06:16 pm - Niels Lindenthal

I like this idea. For example I can imagine that we take care of the custom fields topic.

In addition I will also prepare an overview on what we are working on until Summer and End-of-Year. This roadmap might reduce the risk that two persons are working on the same feature and don't know about it.

#### 2011-02-01 10:49 pm - Felix Schäfer

- *Status set to Open*

**2011-02-18 07:33 am - Felix Schäfer**

As discussed in #196: upgrade rails to 2.3-latest. Official and tested support for ruby 1.9 would be great too.

**2011-02-18 10:25 am - Pedro Gutierrez**

As a merely involved chicken user I'd be happy to see:

- \* tagging
- \* and/or more custom field types, specially with associated data as proposed by Eric (list of users and list of projects)

**2011-02-22 08:17 pm - Gregor Schmidt**

As just discussed on IRC:

Our idea (finnlabs) was to make the home page and project overview page fully customizable. This would include

- \* implement all currently used 'widgets' like latest projects, member list, and latest news as wiki macro
- \* remove the currently used views and replace them (potentially) special wiki pages
- \* add a way to change these wiki pages and make sure, that permissions are handled correctly

edavis10 also proposed to customize the MyPage in a similar way

- \* implement all currently available widgets as wiki macro
- \* replace the current my page custom view with a personal wiki page

Please correct me, if I missed something or got something wrong.

**2011-02-22 08:32 pm - Felix Schäfer**

I object to having it so closely coupled to the wiki as one could imagine it to be when reading the precedent note. Having most or all of them available as wiki macros: I'm all for it, implementing them as wiki macros: no.

I had a look at "cells":<http://cells.rubyforge.org/> (touted as "components for rails") which essentially provide beefed up partials that support caching and other nice things, and I think they are really promising and could replace a lot of things: the "my page" blocks, the simple issue lists (my page blocks, subtasks, roadmap, etc), maybe a global sidebar, and so on. I'm not sure how feasible it would be to incorporate them in wiki macros though as they're typically called in views rather than "dumped" into strings passed to views.

**2011-02-23 12:07 am - Eric Davis**

Bit of changes for Gregor Schmidt's note:

- \* "remove the currently used views and replace them (potentially) special wiki pages" => not special "wiki pages" but text formatted sections (ie. not @WikiPage@ but @textilizable@ inside a view)
- \* "replace the current my page custom view with a personal wiki page" => with a set of personal my page blocks like how it works now (top, right, left, etc) but can be edited to store wiki content. When added they would just have the Wiki macro be default. Example:

```
<pre>
Add my page block for "Reported issues" to right => Add {{reported_issues}} to a new section on the right side
</pre>
```

Felix:

Feel free to look into cells. My only concern is that many "components" that aren't the "Rails way" end up abandoned, unmaintained, or removed (e.g. Rails 1.x components, Rails Engines). The main points for me are: easy for a user to use, easy to customize, safe to use (XSS), and less code for use to maintain.

(We should probably take the discussion of blocks, wiki macros, and cells into the forums once there are more details to discuss)

**2011-02-23 12:18 am - Andrea Campi**

Felix Schäfer wrote:

> I had a look at "cells":<http://cells.rubyforge.org/> (touted as "components for rails") which essentially provide beefed up partials that support caching and other nice things, and I think they are really promising and could replace a lot of things: the "my page" blocks, the simple issue lists (my page blocks, subtasks, roadmap, etc), maybe a global sidebar, and so on. I'm not sure how feasible it would be to incorporate them in wiki macros though as they're typically called in views rather than "dumped" into strings passed to views.

My experience with Cells is that it's a very interesting and very promising approach, but it's in a bit of "in between" state on Rails 2.3; most of the development and especially documentation is focused on Rails 3 nowadays.

You should still be able to get relatively simple functionality in a reasonable amount of time, but going past that (for example, self-contained widgets that take care of the server-side portion of Ajax calls) may prove a bit of a challenge.

Still worth investigating, just keep that in mind :)

That said: I love the basic idea of using the wiki as the home page / my page.

**2011-03-02 11:02 am - Gregor Schmidt**

According to my comments on #133, I would like to propose to rethink the scope of 2.0.

\*My proposal:\*

Make 2.0 the 'dependency release'! We would focus on getting all external and pseudo-external dependencies up-to-date. This would of course include Rails #196, but also other things like RedCloth, CodeRay, ruby-tree and what not. We should also consider removing all bundled libraries and adding explicit external dependencies.

We would then keep feature development back for another release cycle.

These updates would need a major release since we would need to drop the support for 1.8.6 b/c rails 2.3.11 does not support 1.8.6 as far as I know.

Furthermore I would recommend to integrate bundler in this 2.0 release to further ease the installation and dependency management.

I also think, that we should make 1.9.2 at least beta-supported.

This release should happen rather sooner than later, i.e. we should skip 1.2.

-----

All these awesome ideas collected in this ticket would then become part of 3.0 or such.

This approach would 1. ease the burden of maintaining hacks to support outdated versions and back-porting bug and security fixes we could otherwise have for free. Also this approach would hopefully minimize installation issues of our users and therefore it would minimize the support work.

**2011-03-03 12:07 am - Eric Davis**

Gregor Schmidt wrote:

> Make 2.0 the 'dependency release'!

I kind of like this idea, with a few caveats.

> This release should happen rather sooner than later, i.e. we should skip 1.2.

I don't think 1.2 (or 1.x) should be skipped but happen until we have 2.0 released. That way people can stay up to date and still switch over to 2.0 later on.

> All these awesome ideas collected in this ticket would then become part of 3.0 or such.

There still are a few things I'd like to put into 2.0 but we might be able to wait. I'd at least like to do some cleanup and finish what I started with subtasks (bugs IMO).

Any other thoughts?

**2011-03-03 12:26 am - Eric Davis**

I'd also like to sync up with Redmine master for 2.0. There is still a lot of activity with the SCM backends.

**2011-03-05 11:53 am - Felix Schäfer**

Eric Davis wrote:

> Any other thoughts?

I think 1.2.0 is "too soon" to skip or do a "version bump"/upgrade release, but seeing that more and more stuff seems to get complicated/strange because of outdated dependencies (net-ldap, rails, redcloth, coderay, etc) I'd favor making 1.3 a major upgrade to get all these things up to speed, then we can either keep summer-ish for the next major, or go on with the 6 months cycle from there.

**2011-03-07 11:03 pm - Holger Just**

We (Eric, Felix, Holger) talked in IRC a bit today. These are our ideas:

- \* Release version#6 as planned on rails 2.3.5
- \* Skip 1.3 and release version#5 instead. It would include
  - \*\* Move to Rails 2.3.11
  - \*\* Drop Ruby 1.8.6 support
  - \*\* externalize RubyTree, CodeRay, RedCloth and others
  - \*\* Upgrade the LDAP gem
- \* Include the S&P theme if possible (but I'd think that's optional)

Feature development would continue after the version#5 release.

**2011-03-07 11:31 pm - Eric Davis**

Holger Just wrote:

> \* Skip 1.3 and release version#5 instead. It would include

Should be "Skip MINOR\_VERSION once we are ready for RC", depending on how long it would take to get everything ready.

- > \*\* Move to Rails 2.3.11
- > \*\* Drop Ruby 1.8.6 support
- > \*\* externalize RubyTree, CodeRay, RedCloth and others
- > \*\* Upgrade the LDAP gem

Who can commit to doing the work for these?

> \* Include the S&P theme if possible (but I'd think that's optional)

I don't know if this will be ready. I have the basics done but need advice on some of the more advanced parts. I'd like to try to include this if possible.

**2011-03-08 06:32 am - Eric Thomas**

Eric Davis wrote:

> Holger Just wrote:

> > \*\* externalize RubyTree, CodeRay, RedCloth and others

>

> Who can commit to doing the work for these?

When you say externalize those gems, you mean defining them in the config section of the environment, correct? If so, I can handle the redcloth and coderay gems. I have already started working towards this, but need to fix some broken tests.

**2011-03-08 07:07 am - Gregor Schmidt**

I like the plan. Thanks for figuring it out. Will these changes happen on unstable or master first?

**2011-03-08 10:54 am - Felix Schäfer**

Eric Thomas wrote:

> When you say externalize those gems, you mean defining them in the config section of the environment, correct?

I think we should seize the opportunity to switch to bundler. Thoughts?

**2011-03-08 10:54 am - Felix Schäfer**

Gregor Schmidt wrote:

> I like the plan. Thanks for figuring it out. Will these changes happen on unstable or master first?

Most if not all on the unstable branch.

**2011-03-08 10:55 am - Felix Schäfer**

Holger Just wrote:

> \*\* externalize RubyTree, CodeRay, RedCloth and others

> \*\* Upgrade the LDAP gem

I'd make the LDAP gem an external dependency (i.e. bundle it) too.

**2011-03-08 11:02 am - Andrea Campi**

Felix Schäfer wrote:

> Eric Thomas wrote:

> > When you say externalize those gems, you mean defining them in the config section of the environment, correct?

>

> I think we should seize the opportunity to switch to bundler. Thoughts?

I think so too.

I'm trying to set aside all of Friday for working on ChilliProject, if that happens I can do some work in this area.

**2011-03-08 12:19 pm - Eric Thomas**

Felix Schäfer wrote:

> Eric Thomas wrote:

> > When you say externalize those gems, you mean defining them in the config section of the environment, correct?

>

> I think we should seize the opportunity to switch to bundler. Thoughts?

Great idea. If a lot of these gems are getting pulled out of the packaging then the process to install them should be trivial. I've always found bundler to

do a better job of resolving dependencies as opposed to @rake gems:install@.

**2011-03-08 12:47 pm - Felix Schäfer**

Felix Schäfer wrote:

> Holger Just wrote:

>> \*\* externalize RubyTree, CodeRay, RedCloth and others

>> \*\* Upgrade the LDAP gem

>

> I'd make the LDAP gem an external dependency (i.e. bundle it) too.

To clarify: I didn't mean "bundle" as in "bundle the gem with ChiliProject" but as in "put it in the Gemfile for bundler to take care of"...

**2011-03-08 12:55 pm - Gregor Schmidt**

I think, its two parts, that can be discussed and performed separately.

# get rid of other people's code in the chiliproject-sources, i.e. @vendor/gems@ and/or @lib/redcloth3.rb@

# introduce bundle to easy gem management and installation

The first one seems to be discussed and just needs to be performed. The second needs some discussion I guess. I think this ticket is probably not the best place. Cyril Mougel created "a forum thread for this topic":<https://www.chiliproject.org/boards/2/topics/189> today. Perhaps we should hop over to the forums and discuss the topic there?

**2011-03-09 12:36 am - Eric Davis**

Gregor Schmidt wrote:

> # get rid of other people's code in the chiliproject-sources, i.e. @vendor/gems@ and/or @lib/redcloth3.rb@

The only think I have to ask before removing @vendor/@ code is: run a @git log vendor/path@ on each one to make sure there aren't any custom changes to the library. I'd hate to have a regression because we externalized the libraries.

> Perhaps we should hop over to the forums and discuss the topic there?

+1 It's difficult to track everything in this issue. I probably should have started the scoping discussion on the Forums in the first place.

**2011-07-04 10:02 pm - Felix Schäfer**

- Status changed from Open to Closed

version#5 being released, I'd say this definitely closes this issue :-)