Subject: Issue tracking...
Posted by mirek on Wed, 12 Jan 2011 20:05:58 GMT

Well, I am late to this as usual (as it was proposed in the past), but now I am definitely ready for some issue tracking system.

I am happy user of Redmine (www.redmine.org) in my job, so I naturally tend to this solution.

However, I am open for other suggestions.

In any case, if it is Redmine:

- one question is whether it is even possible to install it on www.ultimatepp.org, but I guess that should be ok

- anyway, perhaps we would rather install it to the 'hidden' infrastructure server?

Do we want to keep issue tracking hidden (so we do not need to watch our tongues that carefuly), or make it public?

Subject: Re: Issue tracking...
Posted by dolik.rce on Wed, 12 Jan 2011 20:14:48 GMT

It would clean up the forum a bit... Why not the tracker that already exists on sf.net? Only reason I can see is that it is not on ultimatepp.org (maybe that might be fixed with a little work).

Honza

Subject: Re: Issue tracking...
Posted by mirek on Wed, 12 Jan 2011 20:20:07 GMT

OK, I shall check sf.net tracker first...

Subject: Re: Issue tracking...
Posted by mirek on Wed, 12 Jan 2011 20:27:19 GMT

And the googlecode one..

Subject: Re: Issue tracking...
Posted by mirek on Wed, 12 Jan 2011 20:35:38 GMT
View Forum Message <> Reply to Message

And to sum it up, I was also considering issue tracking as nice case-study for U++ based web development:)

Subject: Re: Issue tracking...
Posted by koldo on Thu, 13 Jan 2011 08:10:51 GMT
View Forum Message <> Reply to Message

Hello Mirek

This is a great idea!.

Subject: Re: Issue tracking...
Posted by mr_ped on Thu, 13 Jan 2011 08:53:32 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 12 January 2011 21:35And to sum it up, I was also considering issue tracking as nice case-study for U++ based web development:)

That would be very interesting, but I think U++ can benefit from tracker usage *now*, and doing such case study in hurry is not really up to my taste. (although considering how the git was born, it looks like it's my own personal problem (of not doing things in a hurry))

Subject: Re: Issue tracking...
Posted by unodgs on Thu, 13 Jan 2011 10:57:10 GMT
View Forum Message <> Reply to Message

Issue tracking is generally a good idea. But I don't need it too much now so if that would speed up developing rainbow project it can be written in upp for me

---

Subject: Re: Issue tracking...
Posted by tojocky on Thu, 13 Jan 2011 14:31:58 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 12 January 2011 22:35And to sum it up, I was also considering issue tracking as nice case-study for U++ based web development:)
I agree with you to make U++ based web development.
I planning to create an issue tracking, integrated with GIT repository (or SVN). Based on web and

windows forms.
Web forms i planning to add with v8cgi.

Finally, I can contribute in this project!

Subject: Re: Issue tracking...
Posted by mirek on Thu, 13 Jan 2011 18:17:43 GMT
View Forum Message <> Reply to Message

unodgs wrote on Thu, 13 January 2011 05:57Issue tracking is generally a good idea. But I don't
need it too much now so if that would speed up developing rainbow project it can be written in upp
for me

Well, that 'rainbow' thing is primarily intended to speed up MacOSX/Android/framebuffer ports.
Using it for TS-like server was just additional idea; I am afraid that in the end we should support
'proper' web development as well...

Subject: Re: Issue tracking...
Posted by tojocky on Fri, 14 Jan 2011 16:06:40 GMT
View Forum Message <> Reply to Message

mirek wrote on Thu, 13 January 2011 20:17unodgs wrote on Thu, 13 January 2011 05:57Issue
tracking is generally a good idea. But I don't need it too much now so if that would speed up
developing rainbow project it can be written in upp for me

Well, that 'rainbow' thing is primarily intended to speed up MacOSX/Android/framebuffer ports.
Using it for TS-like server was just additional idea; I am afraid that in the end we should support
'proper' web development as well...

I do not know details 'rainbow'.

Will be exists possibility to:
1. change source code runtime (like in PHP or v8cgi)

This rainbow will not overload the system for more quantity of users?

Subject: Re: Issue tracking...
Posted by mr_ped on Fri, 14 Jan 2011 19:08:12 GMT
View Forum Message <> Reply to Message

PHP can't change code dynamically.

Well, it can by calling "eval()", but that's not used.
(anyway, self modifying code was common practice back in past, and it died for quite a valid reasons)

What you probably have on mind is simple deployment of new changes, but that's just implementation detail and can be solved up to some degree even with C++, and also for larger projects this is absolutely non issue, because you definitely don't want your project to just change suddenly, that requires much more planning and strategy with large projects to keep the app running without inconsistencies and noticeable downtime.

If you want just to do some small page and experiment a lot, PHP is excellent choice and I don't see a reason to compete with it in such case.

Rainbow project is anyway almost completely unrelated to web development, except the ability to run more of U++ core in server environment without GUI and with smaller platform dependency, so porting and maintaining will be simpler.

Subject: Re: Issue tracking...
Posted by chickenk on Wed, 19 Jan 2011 10:27:07 GMT
View Forum Message <> Reply to Message

mirek wrote on Wed, 12 January 2011 21:27And the googlecode one..

Well, after having used it a bit more on other projects, I must say that googlecode issue tracker is not that bad. It seems too 'light' at first when you see the plain text box for entering the bug description, but after all it's sufficient. And the features are elsewhere :

- you (core devs) can apply labels to filter issues by themes (think gmail-like filters)... handy
- any observer can easily subscribe and get email notifications when the issue is updated (right, this one is present on most of issue tracking systems)
- any observer can suggest a patch by directly editing source code from the browser app ! I just used this and it's very handy. It creates an issue of type 'Patch' with the suggested patch attached.
- there are interactions between source code repo and issue list (Redmine does the same kind of thing) with an API that enables to control issues from commit messages (ex. you write 'Fixes issue 1234) and the issue 1234 is changed to fixed.

All in all, it seems everything useful is inside. And I've not seen the owner/admin side of googlecode yet.

My 2 cents,
Lionel

Subject: Re: Issue tracking...
Posted by harmac on Wed, 19 Jan 2011 21:49:25 GMT

When thinking about improvements for development, it might also be a good idea to consider changing to a distributed version control system. There has already been some discussion about that in the coffee corner. I think, I should go there later and also add something to the discussion, as I've recently read about a couple of systems around of which Bazaar and Fossil looked the most promising to me.

Speaking of the latter, it has some unique features and its being BSD licensed might also be worth taking a look on it, as you mentioned having considered issue tracking as a case study for U++ web development (what exactly does that mean, did you think of writing an issue tracker?). In particular, it is self-contained and integrates bug tracking and a wiki for documentation with its version control that allows for distributed working. So, if I've read that correctly, you can for example write offline in the wiki and commit changes locally and push them later to a central repository the same way as you'd do it with code when using the system with autosync off. While a GUI is missing, which I consider a weakness, the author argues that it is not necessary, because there's a web interface.

Otherwise, I'd suggest not to use sourceforge, because I have the impression that the site loads annoyingly slowly when compared to googlecode, which might be the best choice among third party web hosted trackers.

Are there any specific requirements for functionality that shouldn't be missing?

Subject: Re: Issue tracking...
Posted by unknown user on Wed, 19 Jan 2011 23:59:35 GMT

harmac wrote on Wed, 19 January 2011 22:49When thinking about improvements for development, it might also be a good idea to consider changing to a distributed version control system. There has already been some discussion about that in the coffee corner. I think, I should go there later and also add something to the discussion, as I've recently read about a couple of systems around of which Bazaar and Fossil looked the most promising to me.
Off-topic:

IMO git fit better with U++ design. One can checkout a branch (change from one branch to other) in same directory so there is no need to add new assembly to TheIde, only reload the files (which TheIde already do) - one reason why i use git with U++.

If i understand correctly fossil repo is one single file? SQLite DB? You have to run "open" command to map the file to a directory and "close" when you done - that doesn't worth the effort IMO fossil should exist only in museum.

Bazaar is ok (a bit slow), but it can't have branches to share same directory (AFAIK), and switch from one to another.

Andrei

Subject: Re: Issue tracking...
Posted by kohait00 on Thu, 20 Jan 2011 08:34:55 GMT
View Forum Message <> Reply to Message

glad to see things going this direction..
Quote:
When thinking about improvements for development, it might also be a good idea to consider
changing to a distributed version control system.

totally agree with it. this makes developping things really smooth. and cleans up the forum. otoh,
having code examples and 'geek upp terminology' (things beginners wouldnt first search for in
upp/and help forum) really helps the learning process.. so totally dismiss them (issue threads)
there would probably make learning for the 'others' (non-developers) more difficult.

in terms of bug tracking: i'd suggest to use a BT that can be integrated with the version control of
choice (see coffe corner: GIT). so i'd suggest 'mantis' which might not seem too fancy but is just
the right choice for most things, and has GIT integration plugins.

but some research revealed: Redmine is just as potent for integrating GIT. and it looks better

so, for mireks sake, i'd suggest to switch over to Redmine/GIT.
(for git, all version history from svn can be imported, i'm having running a git mirror from upp
locally, and someone is running a public one as well, Mindtraveller, right?). this would speed up
migrating (just pass over a rar file).

Subject: Re: Issue tracking...
Posted by chickenk on Thu, 20 Jan 2011 09:41:15 GMT
View Forum Message <> Reply to Message

I've been suggesting and using Redmine at work for a few projects, and I've always been happy
with it indeed. Seems my colleagues appreciated is as well, they can't do without Redmine now.
So for git and project management integration (including BT and source code viewer), I vote for
Redmine definitely.

Lionel

Subject: Re: Issue tracking...
Posted by mirek on Thu, 20 Jan 2011 13:19:09 GMT
View Forum Message <> Reply to Message

Well, I have found that redmine is in the main Ubuntu 10.04 repo, which seems to make me prefer
'redmine on infrastructure server' variant.

As for GIT, well, I guess at the moment I am too invested into svn. I believe that most of what GIT
would give as can be quite painlessly achieved by making 'branches' directory of svn r/w for every

'primary svn' user.

Subject: Re: Issue tracking...
Posted by kohait00 on Thu, 20 Jan 2011 14:27:03 GMT
View Forum Message <> Reply to Message

sure. no doubt about that.

we are using git in company now for about 1 year. it's been a pain relief . cause branching is really fast and one can keep an eye on the changes that happened. svn is just too slow in that sense. another advantage is the possibility to follow others' repo's and cherry picking some commit while rejecting/ignoring others. nead features that just make life more productive. i can provide some really good information sources on git. maybe reconsider it maybe it's not bad to change/introduce both at once, as one step, to spare some pain.

Subject: Re: Issue tracking...
Posted by fudadmin on Thu, 20 Jan 2011 14:54:53 GMT
View Forum Message <> Reply to Message

harmac wrote on Wed, 19 January 2011 21:49When thinking about improvements for development, it might also be a good idea to consider changing to a distributed version control system.
+1.

Btw, fossil is a good discovery (until someone rewrites UVS ...   )
Fossil Versus Git
GIT ----------------   vs   FOSSIL
File versioning only ----|| Versioning, Tickets, Wiki, and Blog/News
Sharding ----------------|| Replicating
Huge community ------|| Road less traveled
Complex ----------------|| Intuitive
Separate web tools ---|| Integrated Web interface
Lots of little tools -----|| Single executable
Pile-of-files repository-|| Single file repository
Uses "rebase" ----------|| Immutable
GPL ---------------------|| BSD

Subject: Re: Issue tracking...
Posted by fudadmin on Thu, 20 Jan 2011 15:10:32 GMT
View Forum Message <> Reply to Message

andreincx wrote on Wed, 19 January 2011 23:59

If i understand correctly fossil repo is one single file? SQLite DB? You have to run "open" command to map the file to a directory and "close" when you done - that doesn't worth the effort IMO fossil should exist only in museum.

Andrei

Then all databases are bad?

Subject: Re: Issue tracking...
Posted by kohait00 on Thu, 20 Jan 2011 15:23:49 GMT
View Forum Message <> Reply to Message

don't want to start a flamin discussion git vs. fossil. haven't tried the latter.

thanks for the comparison. though, not all points, that seem 'negative' or 'disadvantage' are really such.

GIT in no way is to complicated. the problem is, that we are subversion-braindamaged and need a paradigm shift but thats true for all distributed vcs. so it depends on perception.

licence: GPL, doesnt bother us, since we wont change or reuse the source code, but use the tool.

separate web tools: we keep the choice 'what else' besides dvcs best fits our needs, which forum, which blog, which bug tracker.

multiple files: is definitely better. shorter load times etc..less mem consumption during run.

etc..everything is relative (to the needs)

Subject: Re: Issue tracking...
Posted by unknown user on Thu, 20 Jan 2011 15:51:53 GMT
View Forum Message <> Reply to Message

fudadmin wrote on Thu, 20 January 2011 16:10andreincx wrote on Wed, 19 January 2011 23:59
If i understand correctly fossil repo is one single file? SQLite DB? You have to run "open" command to map the file to a directory and "close" when you done - that doesn't worth the effort IMO fossil should exist only in museum.

Andrei

Then all databases are bad?

Databases are not bad, but VCS based on databases are bad (if they don't hide that dependency), i personally dislike monotone and now fossile maybe it's just me.

Andrei

Subject: Re: Issue tracking...
Posted by dolik.rce on Fri, 28 Jan 2011 13:28:01 GMT
View Forum Message <> Reply to Message

Back to the topic of issue tracking

I am quite new to this, so I would like to make sure I get some things right. So my idea about typical flow of things is this:

1) Report a bug/task/feature. It's state is "New".
2) Someone makes a decision if the issue is worthed the work/necessary/good idea and sets it to "Approved".
3) Someone assigns the issue to themselves and starts working on it. The state changes to "In Progress".
4) Assigned person (optionally?) reports progress.
5) When it is ready, the issue state is changed to "Ready for QA" and is reassigned to someone responsible who checks if it is OK.
6) Issue is either "Closed" and changes committed or it is assigned back to the person who worked on it, to continue at step 4) with state "In Progress".
Few consecutive steps can be possibly merged.

Is that correct? Few things are still unclear to me, though in most cases the "common sense solution" is probably correct. Like who can approve an issue (can I approve issue I reported?) or who to ask for QA (Mirek in case of new features, original creator of changed code in other cases?). Are there any other rules we should follow?

Honza

Subject: Re: Issue tracking...
Posted by Novo on Sat, 29 Jan 2011 15:30:54 GMT
View Forum Message <> Reply to Message

off-topic:

I'd like to suggest to use buildbot for automated building and testing. It is a distributed system. A master process can run on the same server as redmine and build slaves can be run somewhere else. For example, I do not have any BSD system installed in my household, but somebody else might have it.

This is a quite useful system and I use it a lot at work.

Subject: Re: Issue tracking...
Posted by mirek on Sat, 29 Jan 2011 19:35:47 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Fri, 28 January 2011 08:28Back to the topic of issue tracking

I am quite new to this, so I would like to make sure I get some things right. So my idea about typical flow of things is this:

New -> Rejected (closed)
    -> In progress (or just can stay New)
        -> Ready for QA
            (usually switched to reporter)
then reporter either Approved (closed) or returns it to New or In progress.

Quote:
Like who can approve an issue (can I approve issue I reported?) or who to ask for QA (Mirek in case of new features, original creator of changed code in other cases?).

IME it is best when one that reported the issue approves it. Of course, not always possible.

E.g. in #19, the task to do is me to check the code and apply it to uppsrc. When I am done, I will switch it back to you for final check and approve.

Mirek

Subject: Re: Issue tracking...
Posted by dolik.rce on Sun, 30 Jan 2011 17:20:36 GMT
View Forum Message <> Reply to Message

mirek wrote on Sat, 29 January 2011 20:35dolik.rce wrote on Fri, 28 January 2011 08:28Back to the topic of issue tracking

I am quite new to this, so I would like to make sure I get some things right. So my idea about typical flow of things is this:

New -> Rejected (closed)
    -> In progress (or just can stay New)
        -> Ready for QA
            (usually switched to reporter)

then reporter either Approved (closed) or returns it to New or In progress.

Quote:
Like who can approve an issue (can I approve issue I reported?) or who to ask for QA (Mirek in case of new features, original creator of changed code in other cases?).

IME it is best when one that reported the issue approves it. Of course, not always possible.

E.g. in #19, the task to do is me to check the code and apply it to uppsrc. When I am done, I will switch it back to you for final check and approve.

Mirek

Ok, that clears it up for me, thanks Mirek. I was mystified about the Approved state - it just didn't occur to me that it means "approve the solution" (not "approve the issue"). My English is getting worse and worse over the time

Honza

Subject: Re: Issue tracking...
Posted by dolik.rce on Mon, 31 Jan 2011 15:25:41 GMT
View Forum Message <> Reply to Message

One more question

I accidentally double-posted (issues 24&25) and found out that there is probably no way to delete/close one of them. Actually not even mark them as duplicate. Shouldn't users with developer status have right to close issues? Or are they closed automatically on Approve? But that wouldn't make sense in this case, as the bug is not approved yet, I just want to "hide it" somehow to prevent the same thing to be discussed in two places.

Honza

Subject: Re: Issue tracking...
Posted by mirek on Mon, 31 Jan 2011 17:52:37 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Mon, 31 January 2011 10:25One more question

I accidentally double-posted (issues 24&25) and found out that there is probably no way to delete/close one of them. Actually not even mark them as duplicate. Shouldn't users with developer status have right to close issues? Or are they closed automatically on Approve? But that wouldn't make sense in this case, as the bug is not approved yet, I just want to "hide it"

somehow to prevent the same thing to be discussed in two places.

Honza

In such cases, "Rejected" is the right option, with possibly mentioning in the text that it is a duplicate of another issue.

(Or you can even add 'related' issue).

Note that such Reject is also appropriate in situation where e.g. new bug report duplicates some older one.

Mirek

Subject: Re: Issue tracking...
Posted by dolik.rce on Mon, 31 Jan 2011 18:46:30 GMT
View Forum Message <> Reply to Message

mirek wrote on Mon, 31 January 2011 18:52dolik.rce wrote on Mon, 31 January 2011 10:25One more question

I accidentally double-posted (issues 24&25) and found out that there is probably no way to delete/close one of them. Actually not even mark them as duplicate. Shouldn't users with developer status have right to close issues? Or are they closed automatically on Approve? But that wouldn't make sense in this case, as the bug is not approved yet, I just want to "hide it" somehow to prevent the same thing to be discussed in two places.

Honza

In such cases, "Rejected" is the right option, with possibly mentioning in the text that it is a duplicate of another issue.

(Or you can even add 'related' issue).

Note that such Reject is also appropriate in situation where e.g. new bug report duplicates some older one.

Mirek
Oups, I overlooked rejected. Does that close the issue too? Also, I didn't see an option to add related or duplicate issue anywhere...

Anyway, thanks a lot for your patience

Honza