

Code Review

As per [item4685](#), this project should be re-activated.

As per [Where](#), this page should be about general improvements to the Code review feature. Coordination about code.tiki.org should happen on the community site: [code.tiki.org](#)

2012-08-08: Work has restarted in trunk with enhancements to [ratings](#) and [PluginProposal](#), and will be part of Tiki10. It permits the Quality Team to vote directly on code.tiki.org with dashboards (commits without enough votes, commits that have been voted down, etc.). There are also [weights when some votes count more than others](#).

2010-06-14: Work [has started](#) and this is part of [Tiki6](#). You can [Dogfood](#) at [code.tiki.org](#) and on your own server with the [software project profile](#)

Background info

It was brought up in [TikiFestStrasbourg](#) that we want a more systematic way to collectively review code commits.

Our vast project relies on [enough eyeballs](#). This means quantity but also diversity. Right now, we don't know if a commit was checked by 6 people and another, by no one. Let's become more efficient.

ImpressCMS has a nifty system. It's a web-based review queue of all commits. Devs can approve, add a comment, ask for rollback, etc. And for everything except approve, an email is triggered to the dev list and the relevant comments are kept in the web-based system. And they connect this to their [Trac](#) instance.

This is quite similar to the informal way we do it now. We tend to reply to the SVN commit mailing list with feedback.

And if you are not comfortable to review a commit, you just move on to the next one and let a dev with more experience in that part of the code handle it.

This ticket/case tracking is very similar to the use case for [GroupMail](#). I think it could make great dogfood.

We could also imagine two approve buttons "the fix/feature works" and "the code is of quality and future-proof". This could be useful in the case a programmer doesn't have sample data to test. Programmer could report that code is "right way of doing it" and "it should work" while a power-user who has the data could confirm that the "fix indeed works" and "hasn't broken anything for me".

People could put a special syntax in the commit message that could close the bug, add a note, etc.

This can save the [Quality Team](#) a lot of time.

Please see [ImpressCMS example](#)



```
svn commit -m "[FIX] Add new Tiki Coffee feature. Solve #2345" tiki-coffee.php
```

This should trigger to close the ticket #2345 and add the note, and ideally, a link to SVN: [example](#)

Questions

- What if one of the servers is not accessible? Will it retry later?

Related

- [Repository Browser](#)
- [Web Commits](#)
- [Version Control Bridge](#)
- [Subversion](#)
- [Distributed revision control](#)
- [How to figure out what revision number causes a bug](#)
- http://en.wikipedia.org/wiki/Code_review
- <http://blog.lphuberdeau.com/wordpress/2009/01/25/adding-collaboration-and-durability-to-code-reviews/>
- http://profiles.tiki.org/Software_Project
- <http://www.mediawiki.org/wiki/Special:Code/MediaWiki> an example
- `doc:Integrator` `dev:integrator`

Other apps

- [Gerrit](#)
- <http://codestriker.sourceforge.net/> inactive as of 2013-03-02
- [Review Board](#)
- <http://code.google.com/p/jupiter-eclipse-plugin/>
- <http://code.google.com/appengine/articles/rienveld.html> <http://code.google.com/p/rienveld/>
- [Google collaborative code review tool \(PHP\)](#)
- <http://www.hammurapi.com/dokuwiki/doku.php/products:hammurapi:start>
- <https://www.ohloh.net/tags/codereview> and https://www.ohloh.net/tags/code_review
- <http://www.mediawiki.org/wiki/Extension:CodeReview>
- [Apache Allura](#)
- <https://blog.assembla.com/AssemblaBlog/tabid/12618/bid/92726/Try-the-new-Subversion-code-review-and-merge-requests.aspx>
- <https://about.gitlab.com/2017/03/17/demo-mastering-code-review-with-gitlab/>

alias

- code.tiki.org