

## Quality Team

For several years, there was a formal quality team that did code reviews on each commit on stable branches. This was quite resource intense as a team of 5 developers would review, and 3 approvals were necessary for a commit to be approved. And there was not a proper code review tool (votes were handled by the commit mailing list). Thus, this effort has ceased and instead the code review is done "wiki way". This team (as it was) no longer exists. Any quality related automated tools will be via the [Continuous Integration Team](#), and manual processes will be part of the regular activities of the [Developers Team](#).

If we were to revive a code review process, it would be much less resource intense. We'd set up a proper tool for voting and the code management. Please see: [Quality Team - How to reduce the workload](#)

There is stub of this in Tiki: [https://profiles.tiki.org/Software\\_Project#Hooks\\_for\\_version\\_control](https://profiles.tiki.org/Software_Project#Hooks_for_version_control)

The information below is kept for posterity.

The [Quality Team](#) aims to (especially in the stable branch) to share experience and minimize the risk of introducing regression bugs or bad coding practices.

Although there is currently no formal code review process for commits, it is currently being done by the [Developers Team](#) on an ongoing basis through the monitoring of the SVN mailing list. A formal code review process has been implemented before but only for commits in the LTS branches, in the form of a Quality Team. This was abandoned when the amount of overhead became unsustainable. There has been talk about getting some kind of code review process up again, but details are uncertain - in order to have a process that works smoothly and does not depend on too small a group of developers. The plan is to wait till after we've switched to GIT from SVN as that would address a number of issues we faced before including merging issues between the various SVN branches. GIT is also a much better tool via "pull requests" to implement a more distributed code review process. See [Distributed revision control](#).

### Release responsibilities

- The [Quality Team](#) reviews commits according to the [Version Lifecycle](#).

### Ongoing responsibilities

- Maintain [code.tiki.org](http://code.tiki.org) (might not be needed anymore if we are on GIT and using something like GITHUB, to be investigated).
- [Release LTS versions](#)

### Task

- Clean up content and move it to proper site according to [Where](#).
  - [Code Review](#)
  - [Quality Team](#)
  - [Quality Team - How to reduce the workload](#)

### Projects

- Implement [Reduction of workload strategies](#)

More details: dev:[Quality Team](#)

### Aliases

- [Code Review](#)