

# Gerrit vs GitLab

GitLab compared to other DevOps tools

## On this page

- [Summary](#)
- [Comments/Anecdotes](#)
- [Resources](#)
- [Comparison](#)

## Summary

Gerrit is a free, web-based team code collaboration tool. Software developers in a team can review each other's modifications on their source code using a Web browser and approve or reject those changes. It integrates closely with Git. It was originally developed as a code review system for Android development. Gerrit includes code diffs, discussions, a and a git server.

In contrast, GitLab provides a git server, with issues and issue tracking, as well as code review, and features to automate the entire DevOps lifecycle.

## Comments/Anecdotes

- From a customer who used Gerrit:
  - Code Review lacked collaboration and communication: commit had 1 single thread, review by commit instead of branch.
  - Wanted to set up forks but Gerrit has no concept of forks
  - Unable to delete projects, creates a cluttered namespace
  - No group ability
  - Not intuitive - additional training required for any more complex workflows

## Resources

- [Gerrit website](#)
- [Gerrit wikipedia](#)

## Comparison

### FEATURES



### Community Supported

The GitLab Community Forum is an active and vibrant place for all GitLab users to share and seek support.



[Visit the GitLab Community Forum](#)

## Image Discussions

Within a commit view or a merge request diff view, and with respect to a specific location of an image, you can have a resolvable discussion. Have multiple discussions specifying different areas of an image.



[Image Discussions](#)

---

## Merge Request Commit Discussions

Comment on a commit within the context of a merge request itself



[Merge Request Commit Discussions](#)

---

## Multiple approvals in code review

In GitLab, to ensure strict code review, you can require a specific number of approvals on a merge request by different users before being able to merge it. You can undo an approval by removing it after the fact.



[Approvals Documentation](#)

---

## Merge request approvals

When a project requires multiple sign-offs, GitLab Enterprise Edition enables you to make sure every merge request is approved by one or more people. Merge request approvals allow you to set the number of necessary approvals and predefine a list of approvers that will need to approve every merge request in a project, and in-turn improve your code's quality.



[Learn more about merge request approvals](#)

---

## Inline commenting and discussion resolution

Code or text review is faster and more effective with inline comments in merge requests. Leave comments and resolve discussions on specific lines of code. In GitLab, Merge Request inline comments are interpreted as a discussion. You can configure your project to only accept merge requests when all discussions are resolved.



[Learn more about resolving discussions](#)

---

## Code Owners

Assign Code Owners to files to indicate the team members responsible for code in your project using a `CODEOWNERS` file.



[Learn more about Code Owners](#)

---

## Code review dashboards

Dashboards with a filterable set of code reviews (could be by project, by user, by branch, by status, or a combination of those). Dashboards includes code review status and links to get to them. This makes it easy to see what is going on with code reviews for a desired subset.



[Code review with GitLab](#)

---

## Contributor agreements

Users can be required to sign one or more contributor agreements before being able to submit a change in a project.



[Read more on the issue](#)

---

## Robot comments

Support for inline comments that are generated by automated third-party systems, for example robot comments can be used to represent the results of code analyzers.



[GitLab merge requests store results](#)

---

## Works with multiple repository types

Supports more than one repository type, such as Git, Subversion, Perforce, CVS, Mercurial.



[Learn about migrating from other SCMs](#)