Crucible vs GitLab

GitLab compared to other DevOps tools

On this page

- Summary
- Resources
- Pricing
- Comparison

Summary

Atlassian Crucible is a collaborative code review application. Like other Atlassian products, Crucible is a Web-based application primarily aimed at enterprise, and certain features that enable peer review of a codebase may be considered enterprise social software. Crucible is particularly tailored to distributed teams, and facilitates asynchronous review and commenting on code. Crucible also integrates with popular source control tools such as Git and Subversion. Crucible is not open source, but customers are allowed to view and modify the code for their own use. (derived from Crucible wikipedia page)

Like Crucible, GitLab provides code review features, and also is optimized to help large (and small) teams work asynchronously. In addition to code review capabilities, GitLab also provides a Git based source code repository, issue tracking and management, CI/CD built-in, security testing, packaging, release, configuration, and monitoring, all within a single application covering the entire DevOps lifecycle.

Resources

- Atlassian Crucible website
- Crucible Wikipedia

Pricing

- Pricing page
- Small teams
    - $10 US - one-time payment, unlimited repos, up to 5 users
- Growing Teams
    - one-time payment, unlimited repos
        - $1,100 US - up to 10 users
        - $1,650 US - up to 25 users
        - $3,030 US - up to 50 users
        - . . .
        - $13,200 US - up to 500 users
        - $17,600 US - up to 2000 users
        - $22,000 US - 2000+ users

Comparison

FEATURES

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