Improving Contribution Guidelines for First Time Contributors Report

FY24-Q1 OKR

Contributor Success & Technical Writing
Expected Benefits

Achieve a higher success rate of first time contributor onboarding

- Expect tutorial will help first time contributors make a merge request
- Expect GDK setup guides to reduce frustration in getting started
- Expect community forks documentation to remove blockers and improve collaboration
- Expect workflow guidelines to teach good practices and lead to faster team member reviews
- Expect non-developers to find contribution opportunities besides coding

Improve Contributor Success KPIs

- Potential UWCC with open merge requests
  - Expect reducing development blockers and frustrations to lead to more submitted MRs
- UWCC (Unique wider community contributors)
  - Expect more MRs using community forks to reach the finish line
- Returning contributors ratio
  - Expect smoother first time contributing experience to increase likelihood of returning
Contributor Survey Feedback | 2022

Documentation

- Contribution documents related to gdk
- Documentation can be improved with examples
- Detailed readme with clear prerequisites

Although everything is well documented, sometimes link to documentation takes to the page with 1000 lines. It's a bit hard to find relevant thing. For example, understanding commit message format and generating changelog was bit hard for me. It may not be necessarily an issue with GitLab but it could be about me, still, wanted to mention.

It would be nice to have more documentation that does not assume potential contributors to already be part of a ruby/rails community. As a GitLab user who wanted to contribute a feature regardless of the tech stack being used at GitLab, I'd have definitely benefited from that.

Contributors given multiple links for “contributing”

- about.gitlab.com/community/contribute/
- docs.gitlab.com/ee/development/
- docs.gitlab.com/
- developer.gitlab.com/

Google search

- Google linking to user docs instead of contributing docs
- No way to tell the difference
New Contributor Journey

Give one link

- [about.gitlab.com/community/contribute](about.gitlab.com/community/contribute)
- First time contributor tutorial front and center
- All contributing areas easy to find
- Connected back to [docs.gitlab.com/ee/development/](docs.gitlab.com/ee/development/)

How will contributors navigate or stumble?

- Let's do the research to find out
- Stop guessing what contributors are doing
- Work with UX Research for real data from contributor journeys
- Iterate improvements along the way
Learnings

What worked...
- Aligning Contributor Success & Technical Writing teams and OKRs
- Fast and quick iterations
- Feedback/MR reviews from community (see tutorial MR)

What didn’t work...
- Contributor Success DRI not having maintainer rights at start of OKR
  - Slowed down MRs
- Deduplicating content across different projects
  - Required multiple MRs and review processes
- Attempting too much criteria for one quarter
  - Documentation was messier and more unorganized than planned for
Next Steps

**Immediate**

- Redesign about.gitlab.com Contribute page with buttons / navigation elements
- Reorganize docs.gitlab.com contribute process navigation
- Rework issue/merge request workflows

**Future Quarter**

- Align with UX Research on RITE study of contributor documentation journey
- Measure analytics on pages and iterate to improve
- Rework Code Review Guidelines
- Add more tutorials/video walkthroughs
- Reach out for community support, even specific trusted contributors
Examples
Contribute Landing Page
Before & After
Intro pulls away from contributing

Contribute to GitLab
The next GitLab Hackathon is just around the corner!

All contributions to GitLab are subject to the DCO or a CLA. More on the GitLab DCO and CLA.

We want to make it as easy as possible for GitLab users to become GitLab contributors, so we created this contribution guide to help you get started. You will find step-by-step instructions to contribute to development, documentation, translation, and design.

Diversity, Inclusion, and Belonging is one of our values. We want to create a welcoming community for every contributor. Please visit our Code of Conduct page to learn more about our commitment to an open and welcoming environment.

Tutorial: Make a GitLab Contribution

Follow this tutorial for contributing to the development of GitLab.

Learn how to configure the GitLab Development Kit, make code updates, push changes to the community fork, and create a merge request.
### Scannable tracks

#### Contribution Guidelines

**Development**
- Add features, fix bugs, help with reviews, and more. You don’t need to know how to write code to contribute.
- More on Contributing to Development
- More on Contributing to Translation

**Documentation**
- From fixing typos to rewriting pages, anyone can contribute to the GitLab documentation.
- More on Contributing to Documentation
- More on Contributing to Translation

**UX Design**
- Help make GitLab easy to use for a diverse group of people. Propose features, fix design bugs, and enhance code.
- More on Contributing to UX Design
- More on Contributing to Translation

**Evangelism**
- Help grow a community where everyone can contribute. Learn about public outreach, blogging, and running community meetups.
- More on Contributing to Evangelism

**Project Templates**
- Create default templates, repositories, and CI/CD pipeline definitions. Help others around the world get started more quickly.
- More on Contributing to Project Templates

### Wordy and distracting descriptions

#### Development
- Whether it’s fixing bugs, adding new features, helping with reviews, GitLab is a great Open Source community for developers from all backgrounds. Many contributors actually started contributing to GitLab development without being familiar with languages like Ruby.
- More on Contributing to Development

#### Documentation
- Contributing to documentation is a great way to get familiar with the GitLab development process and to meet reviewers and other community members. From fixing typos to better organizing our documentation, you will find many areas where you can contribute.
- More on Contributing to Documentation

#### Translation
- GitLab is being translated into more than 35 languages, and this is also driven by wider community members. If you speak another language, you can join more than 1,500 community members who are helping translate GitLab. Come and help us speak your language!
- More on Contributing to Translation

#### Engagement
- Help us craft a fun and rewarding community experience, and help us spread the word about GitLab to reach more users and contributors. Help us foster a community where everyone feels like they belong, and where everyone is able to contribute. Whether you enjoy thinking about community strategy, blogging, organizing events, doing outreach, or just helping to create a fun and welcoming environment, we could use your help.
- More on Contributing to Engagement

#### Project Templates
- Project templates allow developers to get started with new languages and frameworks quickly. We welcome contribution from individuals or communities to the default templates which ship directly with GitLab. You can help promote your favorite languages and frameworks by helping developers around the world quickly get started not only creating default repositories, but providing them default CI/CD pipeline definitions as well.
- More on Contributing to Project Templates
Links go to subpage instead of docs

Before

Contribute Landing Page

After

Links go directly to docs
Graph misleads low contributor totals

Single metric shows total contributors

Community Contribution Metrics

You can see the wider community contribution data in our community dashboard as well as the number of wider community contributions per milestone.

4,039+ Contributors
Contribute Documentation

Before & After
Before

Contribute Documentation

Clean main page with tracks

Overwhelming main page
Before
Contribute Documentation
After

65+ subpages in navigation

9 direct subpages in navigation

Contribute to GitLab development

Contribute to GitLab

Changelog entries
Community roles
Built-in project templates
Design and UI
GitLab Development Kit
Issues workflow
Merge request concepts
Merge request workflow
Code review guidelines
Style guides
Approval rules development guide
Architecture
Audit Event development guide
CI/CD development
Auto DevOps development
Code intelligence
Code Sandbox development environment
Danger bot
Database development
Deprecation guidelines

Developer guide to logging
Development Rake tasks
Documentation
Distributed tracing
Experiments
Feature flags for GitLab development
Framework - DeclarativePolicy
Frontend development
GitLab Pages development
Geo development
Git LFS
Gitalk development
GitLab Design System
GitLab development style guides
GitLab group migration
GitLab project pipelines
GraphQL development
Import/Export
Integrations
Internal API
Internal users

Issuable-like Rails models utilities
Issue types
Jenkins in local environments
Jira development environment
Kubernetes integration
Observability for stage groups
Package development
Permissions
Product Qualified Lead (PQL) guide
Real-time features
Secure partner onboarding process
Sidekiq development
Spam protection and CAPTCHA
Testing standards and styles
Testing (end to end)
Translate GitLab
Snippets
Service Ping guide
Uploads
Value Stream Analytics
Wikis
Work items and work item tuples

Contribute to GitLab

Tutorial: Make a GitLab contribution
Architecture
Development Rake tasks
Development processes
Development style guides
Feature development
User experience
GitLab Development Kit
GitLab project pipelines
Contribute to GitLab Runner
Contribute to GitLab Pages
Contribute to GitLab Distribution
Contribute to GitLab Design System
Contribute to documentation
Feature flags in the development of GitLab

NOTE: The documentation below covers feature flags used by GitLab to deploy its own features, which is not the same as the feature flags offered as part of the product.

This document provides guidelines on how to use feature flags for the development of GitLab to conditionally and/or incrementally enable features and test them in production/staging.

⚠️ All newly-introduced feature flags should be disabled by default.

Feature flags in the development of GitLab

💡 This document explains how to contribute to the development of the GitLab product. If you want to use feature flags to show and hide functionality in your own applications, view this feature flags information instead.

⚠️ All newly-introduced feature flags should be disabled by default.
Before

Contribute Documentation

Messy development page

● Outdated workflows
● Step-by-step guide links to landing page

Contribute to GitLab

Thank you for your interest in contributing to GitLab. This guide details how to contribute to GitLab in a way that is easy for everyone.

For a first-time step-by-step guide to the contribution process, see our Contributing to GitLab page.

Looking for something to work on? See the How to contribute section for more information.

GitLab comes in two flavors:
- GitLab Community Edition (CE), our free and open source edition.
- GitLab Enterprise Edition (EE), which is our commercial edition.

Throughout this guide you will see references to CE and EE for abbreviation.

To get an overview of GitLab community membership, including those that would review or merge your contributions, visit the community rules page.

If you want to know how the GitLab core team operates, see the GitLab contributing process.

GitLab Inc engineers should refer to the engineering workflow document.

On this page
- Security vulnerability disclosure
- Code of conduct
- Closing policy for issues and merge requests
- Contribution flow
- How to contribute
- GitLab Development Kit
- Contributing changelog
- Issues workflow
- Merge requests workflow
- Style guides
- Implement design & UI elements
- Contribution documentation
- Getting an Enterprise Edition license

After

Streamlined development page

● Updated contribution workflow
● Step-by-step guide links to new tutorial

Contribute to GitLab development

Thank you for your interest in contributing to GitLab. This guide details how to contribute to the development of GitLab.

For a first-time step-by-step guide, see Tutorial: Make a GitLab contribution.

How to contribute

1. Read the code of conduct.
2. Choose or create an issue to work on.
3. Set up the GitLab Development Kit.
4. Open your merge request.

Your merge request is triaged, reviewed, and can then be incorporated into the product.
Before Contribute Documentation After

No guide for first time contributors New tutorial for first time contributors

- GDK walkthrough with pictures
- How to make code changes
- How to update documentation
- Using the community forks
- Getting MR reviewed or finding help

### Tutorial: Make a GitLab contribution

Anyone can contribute to the development of GitLab.

Maybe you want to add functionality that you feel is missing. Or maybe you noticed some UI text that you want to improve.

This tutorial will walk you through the process of updating UI text and related files by using the GitLab Development Kit and the GitLab community fork. You can follow this example exactly to familiarize yourself with the process, or you can choose other UI text to update.

### Steps

To make a contribution, you will:

1. Configure the GitLab Development Kit
2. Make your code updates
3. Push your changes to the community fork
4. Create a merge request

---

**On this page**

- Steps
- Prerequisites
- Step 1: Configure the GitLab Development Kit
  - A new GitLab installation
  - An existing GitLab installation
- Step 2: Change the code
  - Update the translation files
  - Update the documentation
- Step 3: Push your changes to the community fork
- Step 4: Create a merge request