

Open UI Repository:

Contribution Guidelines

January 2025

Open UI Repository – Contribution Guidelines

National Association of State Workforce Agencies (NASWA)

[www.naswa.org/open-ui-initiative](http://www.naswa.org/open-ui-initiative)

Open UI Repository - Contribution Guidelines

# Overview

The Open UI repository serves as our central document management system. It utilizes the GitHub Enterprise cloud platform and has public and private elements. These guidelines outline how to effectively contribute, comment, and collaborate on documents within the Open UI GitHub environment.

## Repository

The repository structure helps organize our documents throughout their lifecycle. Understanding this structure helps you locate resources and submit comments and changes to the right location. It contains the main repository, plus three sub-folders for various operational functions.

* Location: <https://github.com/NASWA-OpenUI/Open-UI-Framework>
* Open-UI-Framework/ # Main repository
  + Overview/ # Reference documents for repository usage
  + Published/ # Approved, final documents
  + Working/ # In-progress documents and drafts

## Basic Concepts

Key terms and processes you'll encounter when using the repository. Familiarity with these concepts will help you navigate GitHub's features effectively.

* **Repository:** The central Open UI storage location for all documents
* **Branch:** A copy of your changes that doesn’t affect the original document
* **Pull Request:** Your request to have changes reviewed and approved
* **Issues:** Discussion topics about specific documents or tasks
* **Comments:** Feedback on documents during review

## Pre-requisites

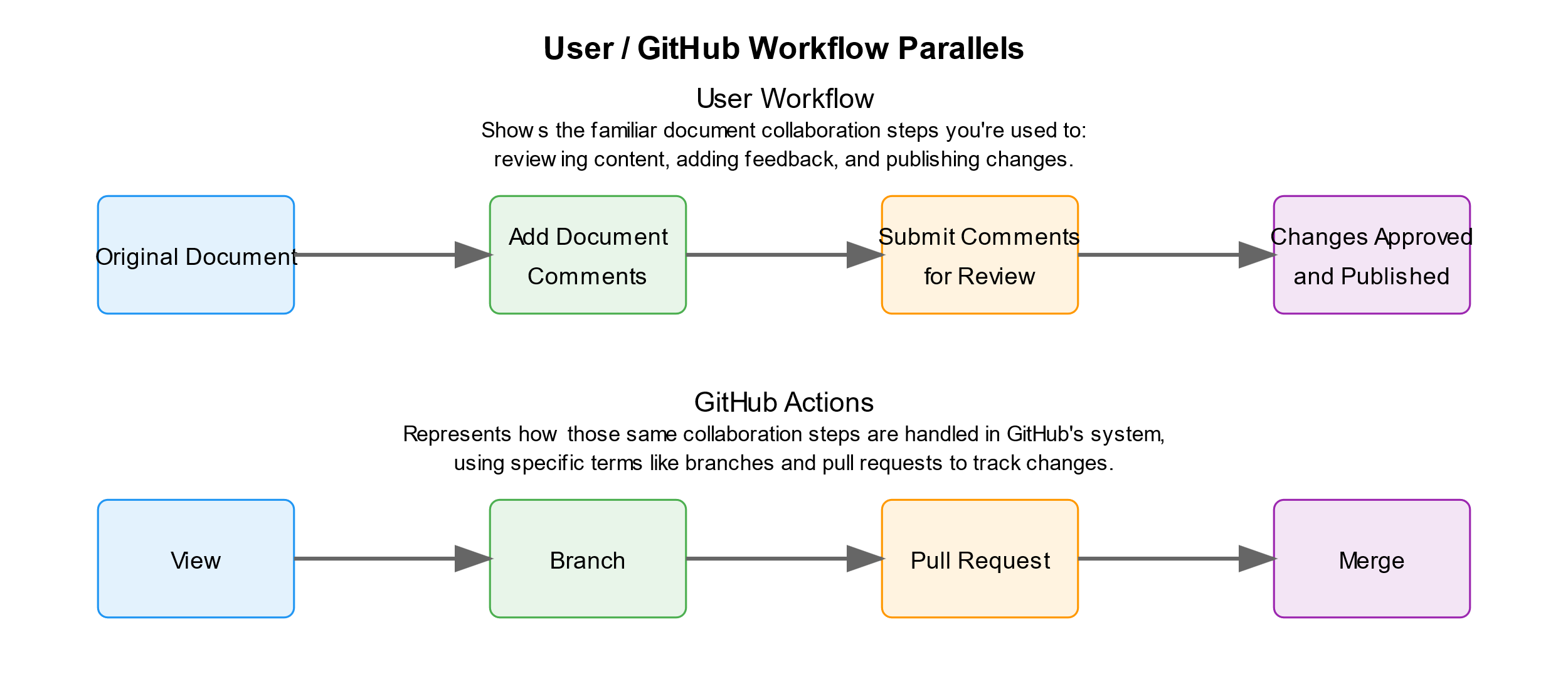
Before accessing the repository, ensure you have:

* Received and accepted your GitHub repository invitation
* Created a new GitHub account or linked your existing account
* Set up Multi-Factor Authentication (MFA)

# Document Management

Learn how to work with documents in the repository - from viewing and finding files to submitting changes for review. While GitHub supports management of raw code files (text), it also supports non-text files such as documents, spreadsheets, pictures, and presentations.

The following diagram shows the correlation between the user actions and the corresponding functions in GitHub:



**Figure 1: User and GitHub Parallel Actions**

## Viewing and Finding Documents

The Open UI repository provides a standard, explorer-type view of the repository’s files and folders. The purpose of this section is to highlight their operational purposes, and how to browse and search for files within them.

* Navigate the repository:
  + Use Published/ for all approved documents
  + Use Working/ for drafts and updates
  + Use Overview/ for guidance and reference materials
* Find specific documents:
  + Browse files in each directory
  + Use the search bar at the top
* Viewing options:
  + For text files:
    - Click file name to preview in browser
    - Click the "Raw" button to view the unformatted text
  + For other files (Word, PDF, etc.):
    - Viewing non-text files is not available through GitHub
    - Click the file name – this will bring up the file preview screen with download options
    - You can either click the text ‘View Raw’, or the download (down arrow) icon
    - Either option will download the file to your computer for editing and comments

## Submitting Comments via Issues

For simple comments and suggestions, it’s recommended that you create an Issue. Non-text files such as Word and .PDF cannot be commented on directly in GitHub, and the Issues feature provides a standardized method to track document discussions, maintain version control, and keep a historical record of events.

* Click the “Issues” button at the top of the screen, and then click “New Issue“
* In the following window:
  + Add a title for the issue
  + Add your narrative in the description window
  + Use the @ notify feature to flag other users (if needed)
  + The options on the right-hand side do not need to be modified (Assignees, Labels, Type, Project, Milestone)
* Click “Create” to submit the issue for review

## Creating a Branch and Submitting a Pull Request

If you have new or edited documents for the repository, you can submit a request for review and approval. There are two steps to the process: 1) creating a branch of your changes, and 2) submitting a pull request to review them.

* Creating a Branch
  + Add documents
    - Navigate to the Working folder
    - Click "Add file" at the top right
    - Choose "Upload files"
    - You can drag and drop files into the browser window or click the ‘choose your files’ link to browse
  + Ensure the option is selected to “Create a new branch for this commit and start a pull request”
    - Name your branch clearly (e.g., "Jan-DOL-report")
    - The additional title and description fields are optional, and are used to populate the automatically generated notifications (defaults to the commit message)
  + Click "Propose changes"
    - You have now created a branch that contains all your changes
* Creating a pull request:
  + After you’ve clicked “Propose changes”, an “Open Pull Request” window will appear
  + Add a descriptive title (defaults to your commit message)
  + Provide additional context in the description:
    - What changed and why
    - Any related documents or issues
    - Use the @ notify feature to flag other users (if needed)
    - Any specific feedback needed
  + Reference any related Issues using # (e.g., "#123")
  + Click "Create pull request" to submit for review

# Review Process and Feedback

The following details are provided for context but are not intended to be prescriptive. The purpose of this section is to provide context and awareness so you may determine how to best participate in the feedback process.

## Track your submission

Monitor the progress of your changes through the review cycle. GitHub provides multiple ways to stay updated and respond to feedback efficiently.

* You’ll be notified of any updates to your open pull requests and issues
* Watch for automatic email notifications
* Monitor both issues and pull requests for reviewer comments
* Respond to any questions or requests

## Provide feedback on documents

Contribute to document improvement through comments and suggestions. Work with reviewers to refine your documents. This collaborative process ensures changes meet quality standards before publication.

* Make requested changes in your branch
* Reply to comments indicating updates
* Request additional review when ready

# Version Management

Track changes, compare versions, and follow document development over time.

## Access previous versions

Locate and retrieve any previous version of a document. GitHub maintains a complete history of all changes for reference and recovery.

* Open any document
* Click the "History" button
* Select any previous version
* Download or view past content

## Compare different versions

Understand how documents evolve over time. GitHub's comparison tools help track changes between versions and identify specific updates.

* Select two versions from history
* See what changed between them
* Review who made specific changes
* Track when changes were made

## Follow document discussions

Keep up with conversations about document development. Following these discussions helps you understand decisions and participate in document evolution.

* View linked pull requests
* Check related issues
* Read comment threads
* See review decisions