

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

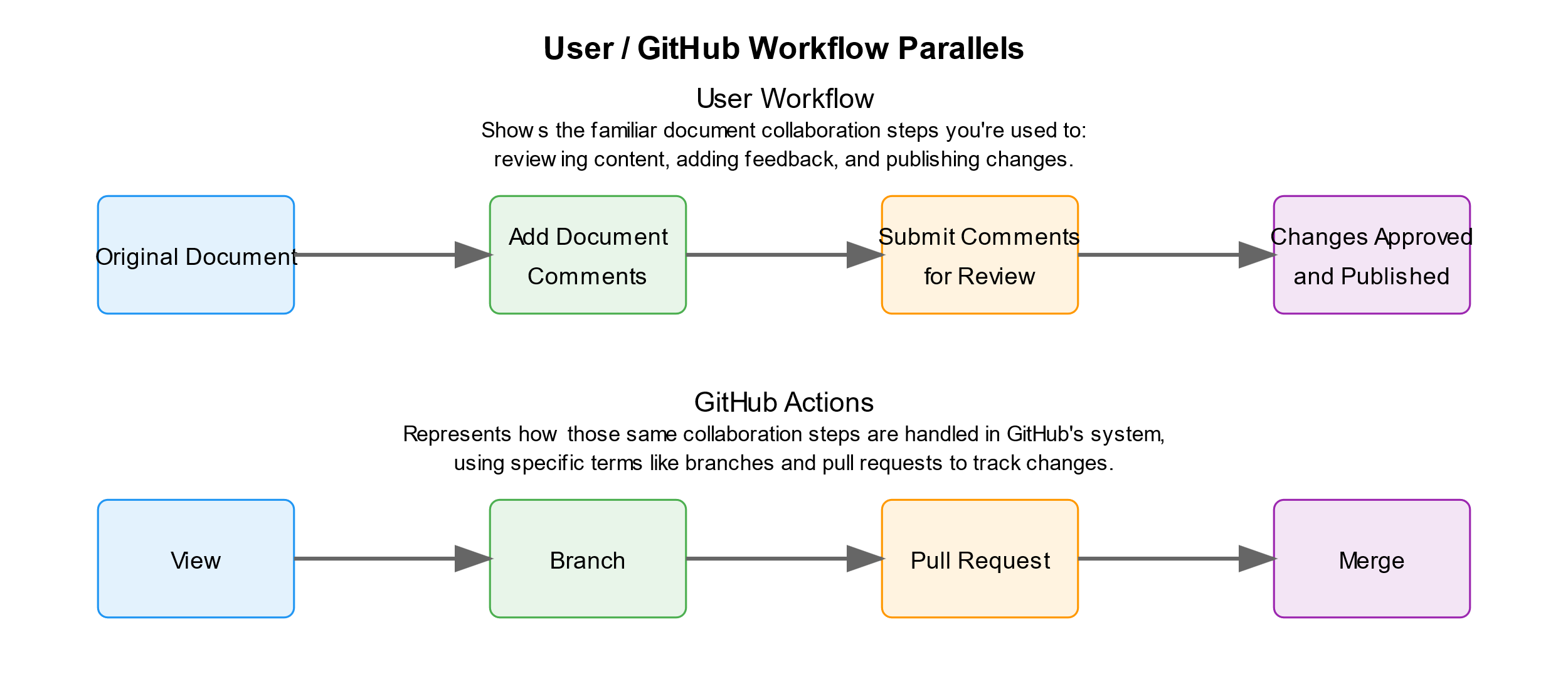
Before accessing the repository, ensure you have:

* 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

Navigate the repository structure to locate documents in their different stages of development.

* Navigate the repository:
  + Use Published/ for approved versions
  + Use Working/ for drafts and updates
  + Use Overview/ for guidance
* Find specific documents:
  + Browse files in each directory
  + Use the search bar at the top
* View and download options:
  + For text files:
    - Click file name to preview in browser
    - Use edit (pencil) icon for changes
    - It you’d like to view the unformatted text (not required), click the "Raw" button
  + For other files (Word, PDF, etc.):
    - Click file name to see download options
    - You can either click the text ‘View Raw’, or the download (down arrow) icon
    - Either option will initiate a download to your computer
    - Viewing non-text files is not available through GitHub

## Working with Documents

Add new content or edit existing documents through different workflows for text and non-text files. All changes are tracked through version control, providing complete history and audit trails.

* Add new documents:
  + Navigate to the Working folder
  + Click "Add file" at the top right
  + Choose "Create new file" for text documents
  + Choose "Upload files" for Word, PDF, etc.
* Update existing documents:
  + Navigate to the document
  + For text files: Click the pencil icon
  + For other files: Upload new version

## Saving and Submitting Changes

Submit your work for review using branches and pull requests. There are two steps to the process, creating a copy of your changes (branch), and submitting the changes for review (pull request).

* When adding or updating documents:
  + Select "Create a new branch"
  + Name your branch clearly (e.g., "Jan-DOL-report")
  + Click "Propose changes"
  + This creates a new branch containing your changes
* On the next screen, create your pull request:
  + Review the changes shown
  + Add a descriptive title (defaults to your commit message)
  + Provide additional context in the description
    - What changed and why
    - Any related documents or issues
    - Who should review (if known)
    - Any specific feedback needed
  + Reference related issues using # (e.g., "#123")
  + Click "Create pull request" to submit for review

# Review Process and Feedback

Understand how document reviews work and how to participate effectively 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.

* Watch for automatic email notifications
* Monitor pull request for reviewer comments
* Respond to any questions or requests

## Provide feedback on documents

Contribute to document improvement through comments and suggestions. Different file types have specific feedback methods to ensure clear communication.

* For text files:
  + Click line numbers to comment on specific content
  + Use pull request comments for general feedback
* For other files (Word, PDF):
  + Add comments in the pull request

**-OR-**

* + Create an Issue and link to the document

## Handling review requests

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