

Open UI GitHub Repository:

Contribution Guidelines

January 2025

Public Repository – Contribution Guidelines

National Association of State Workforce Agencies (NASWA)

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

# GitHub Enterprise Document Repository - Contribution Guidelines

## Overview

This repository serves as our central document management system. These guidelines outline how to effectively contribute, comment, and collaborate on documents within our GitHub Enterprise environment.

## Repository Structure

* **/Overview**  # Reference documents for repository usage
* **/Open-UI-Framework** # Approved, final documents
* **/Working**  # In-progress documents and drafts

## Basic Concepts

* **Repository**: The central Open UI storage location for all documents
* **Branch:** A separate workspace for making changes
* **Pull Request:** The method to propose changes to documents
* **Issues:** Used for discussions and tracking document-related tasks
* **Comments:** Feedback on documents or pull requests

## Contributing Documents

### Adding New Documents

* Navigate to the appropriate directory in GitHub:
  + **/Working** for drafts
  + **/Open-UI-Framework** for final versions
* Click the "Add file" dropdown button at the top of the file list
* Options include:
  + "Create new file" – use this option to create a next text or Markdown file
  + "Upload files" – upload a binary file such as a .PDF, Word document, images, etc
* Do I want to add more details on creating text files in GitHub?

### Before saving:

* Scroll to the bottom of the page
* Select "Create a new branch for this commit"
* Name your branch descriptively (e.g., "add-q1-marketing-plan")
* Click "Propose changes"
* On the next screen, click "Create pull request" to start the review process

### Document Naming Conventions

* We had some general discussion about how to structure the names – for instance, pre-pending working docs with ‘DRAFT’ – and need to regroup with team/confirm naming preferences (if any)

### Commenting and Feedback

* Text Documents (e.g., Markdown, txt)
  + Navigate to the file in GitHub
  + Use line-specific comments by clicking the line number
  + For general feedback, scroll to the bottom of the file
  + Submit your comment
* Binary Documents (e.g., PDF, DOCX)
  + Download the document
  + Add comments using your preferred software
  + Create an issue:
* Link to the original document
* Upload your annotated version
* Provide a summary of changes
* Reference the issue in a pull request if changes are needed

Working with Branches

Branch Naming

Feature branches: feature/document-name

Review branches: review/document-name

Publication branches: publish/document-name

Branch Workflow

Create feature branch from main

Make changes in working/ directory

Review and approval process

Merge to main

Move to published/ when final

Pull Request Process

Create pull request from your branch to main

Fill out the pull request template:

Document purpose

Changes made

Reviewers needed

Request reviews from relevant team members

Address feedback

Merge once approved

Best Practices

Document Management

Keep files organized in appropriate directories

Use clear, descriptive filenames

Include document metadata where possible

Maintain version history in commit messages

Collaboration

Check for existing discussions before creating new ones

Use clear, constructive language in comments

Reference related issues/pull requests

Mention specific team members using @username

Version Control

Make atomic, focused changes

Write clear commit messages

Keep pull requests focused on single documents/topics

Use branches for work in progress

Permission Levels and Access Control

Repository Roles and Permissions

Admin

Full repository control

Manage repository settings

Delete repository

Manage security settings

Configure protected branches

Manage all access levels

Maintain

Push to protected branches

Manage pull requests

Merge pull requests

Configure repository settings (except security)

Create and edit releases

Write

Clone and push to repository

Create and manage branches

Submit and review pull requests

Create and edit releases

Create and manage project boards

Triage

Clone repository

Create and manage issues

Comment on pull requests

Label and milestone management

Cannot write to repository

Read

Clone repository

View code

View issues

Create issues

Fork repository

Cannot write to repository

Common Role Mappings

Document Owners: Admin or Maintain

Content Contributors: Write

Reviewers: Write or Triage

Stakeholders: Read

External Collaborators: Read or Triage

Protected Resources

published/ directory: Protected branch rules

Require pull request reviews

Restrict direct pushes

Require status checks

working/ directory: Standard branch rules

Allow direct contributions

Optional review requirements

Branch Protection Rules

Main Branch

Require pull request reviews

Require approvals (minimum 1-2)

Enforce linear history

Protect against force push

Published Directory

Additional review requirements

Status checks required

Stricter merge criteria

Working Directory

Flexible contribution rules

Optional review requirements

Team-based access controls

Getting Help

Contact repository administrators: [List of admins]

Reference our internal GitHub guides: [Link to guides]

Join our Slack channel: #github-help

Future Considerations

These guidelines will evolve as we expand our GitHub usage to include:

Code management

DevOps integration

CI/CD workflows

Automated document processingmature.