

17.2 Branching Approach

[Go To Top](#)

[Return to Git and Review Workflow](#)

The team should use a simple branching approach for Phase 0 unless the project requires a more formal release model. The branching approach should support parallel development, review, Baseline stability, and demonstration preparation without creating unnecessary process overhead.

A practical Phase 0 branching model may include:

1. A protected main branch for accepted Baseline content.
2. Short-lived feature branches for specific changes.
3. Review branches for larger coordinated updates.
4. Release or demonstration branches where the team needs to freeze a specific Baseline.
5. Tags for accepted Phase 0 Baselines, release candidates, or demonstration runs.

Developers should create branches for changes that affect source files, IDL definitions, generated Artifact policy, scripts, Configuration, containers, tests, documentation, or acceptance Evidence. Developers should avoid committing directly to the protected Baseline branch unless the team explicitly permits it.

Branch names should identify the purpose of the change. Representative branch names include:

- feature/control-plane-status-topic
- feature/generate-types-script
- fix/container-networking
- docs/node-catalogue
- baseline/phase0-demo

The team should avoid branch names that include personal notes, unclear abbreviations, temporary labels, or editing-history terms such as new, old, final, or backup.

© 2026 Dido Solutions, Inc. and Jackrabbit Consulting, Inc.

From:
<https://wiki.didosolutions.com/> - **Dido Solutions, Inc Wiki**

Permanent link:
<https://wiki.didosolutions.com/fxdemo/05-part/17-git-and-review-workflow/17-2-branching-approach/start>

Last update: **2026/08/12 15:43**

