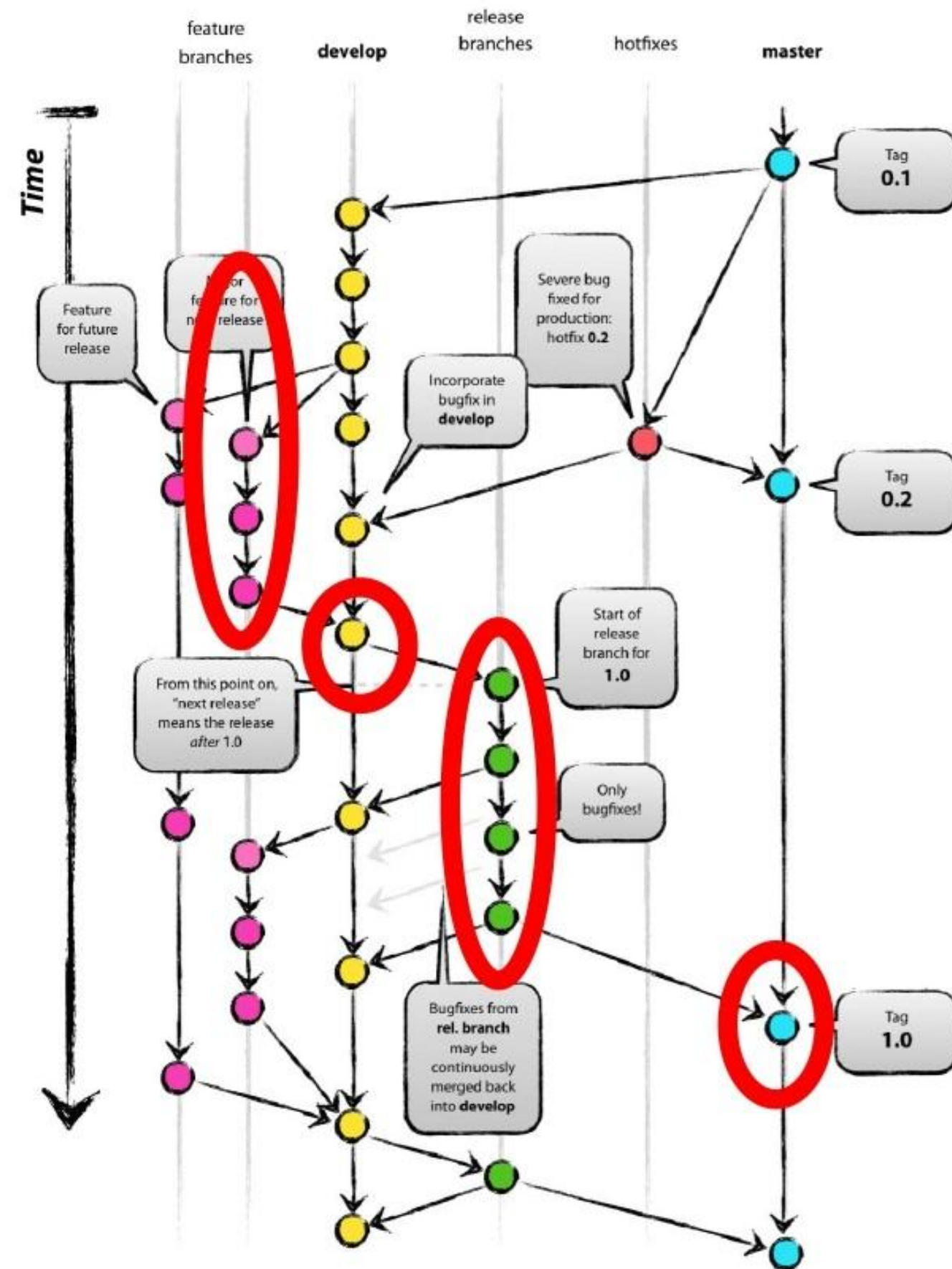


Branching Model

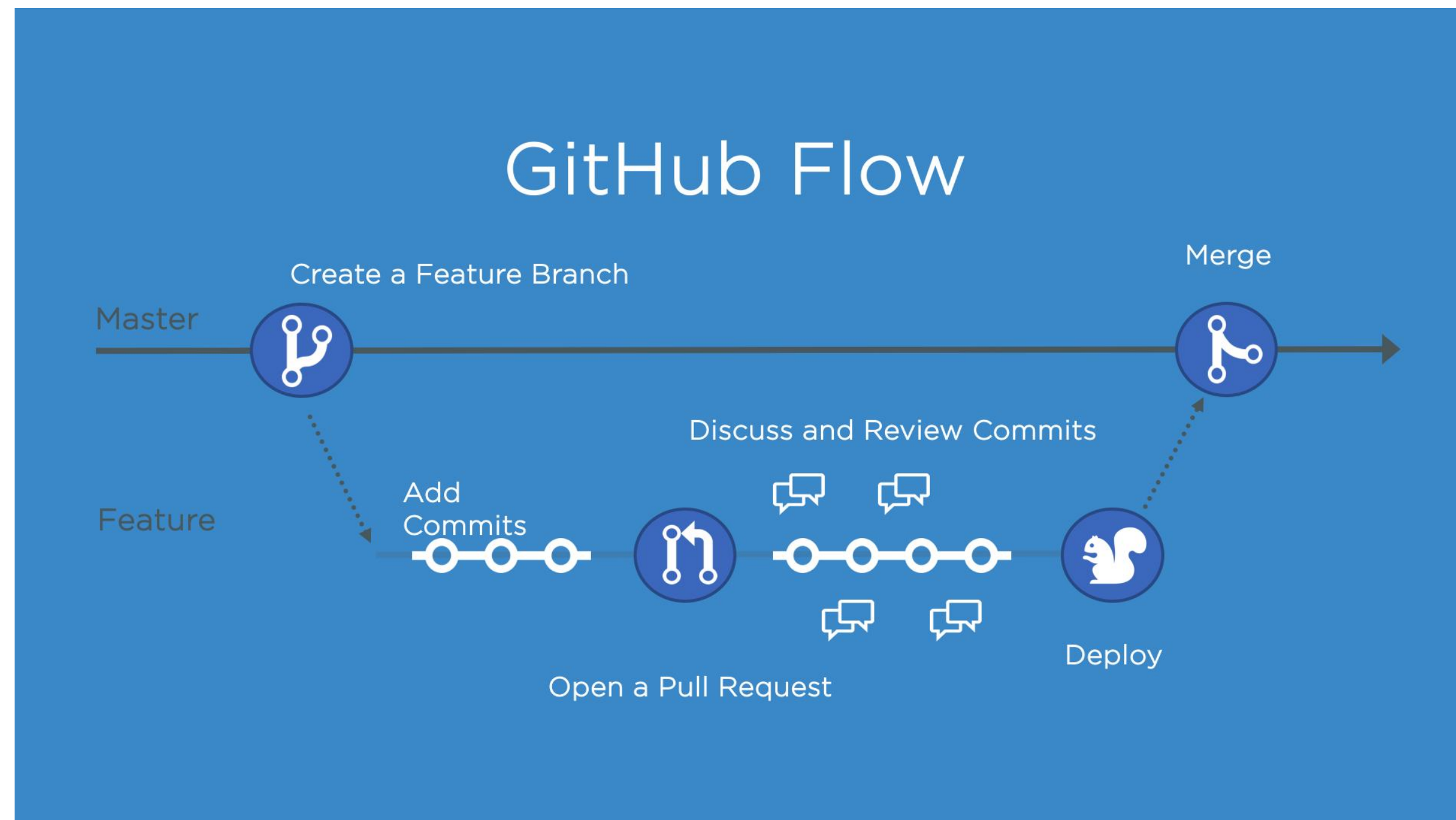
Git flow overview



Workflow Summary:

- 1. Feature Development:** Work is done on separate feature branches created from the develop branch.
- 2. Integration:** Features are merged into the develop branch.
- 3. Release:** Once enough features are developed, a release branch is created for testing and bug fixes.
- 4. Finalization:** The release branch is merged into the main branch and tagged with a version number.
- 5. Hotfixes:** Emergency fixes are developed in hotfix branches directly off the main branch and merged back into both main and develop.

GitHub flow



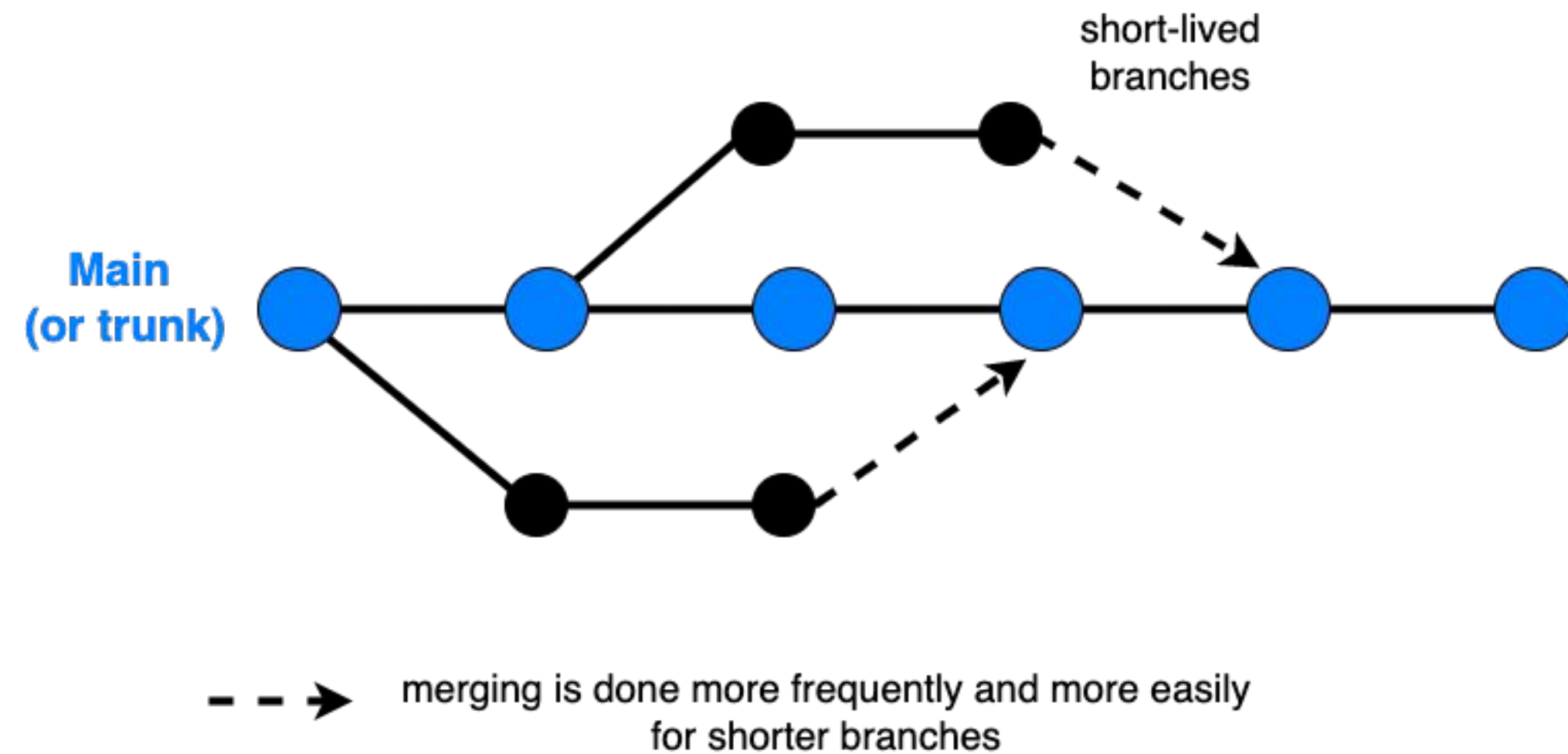
Workflow Summary:

- 1. Create a Branch:** Start by creating a new feature branch from the main branch.
- 2. Work on the Feature:** Develop the feature or task in the new branch, committing changes regularly.
- 3. Open a Pull Request (PR):** Once the feature is complete, open a PR to merge the feature branch into the main branch.
- 4. Review and Testing:** Have the code reviewed by teammates. Automated tests (CI) run to ensure nothing is broken.
- 5. Merge:** After approval and passing tests, merge the PR into the main branch.
- 6. Deploy:** The code is immediately deployable, and, if continuous deployment is enabled, it can be automatically deployed.

Trunk-based development

Trunk-based development

StatusNeo



Feature:

1. No develop branches
2. All implementation occurs directly on the main(trunk) branch

Comparison

Criteria	Git Flow	GitHub Flow	Trunk-Based Development
Main Branch	Uses <code>main</code> and <code>develop</code> branches	Only <code>main</code> branch	Only <code>main</code> (trunk) branch
Feature Branches	Long-lived, separate feature branches	Short-lived feature branches	Very short-lived or direct commits to <code>main</code>
Merging	Merge to <code>develop</code> first, then <code>main</code>	Merge directly into <code>main</code> via PRs	Frequent merges directly into <code>main</code>
Deployment Frequency	Less frequent, after major releases	Frequent, after every merge	Continuous deployment, very frequent
Complexity	More complex, with multiple branches	Simpler, fewer branches	Simplest, directly from <code>main</code>

trunk-based deep dive

simple demo: https://www.youtube.com/watch?v=7_mOxnUCUJA

Advantages:

- 1. Not feature based.**
- 2. Fast collaboration**



**Thank you for
your attention!**